Visual Basic 6
תכנות משחקי מחשב

בנספח תמצא את הוראות
ההתקנה וההפעלה של התקליטור

עריכה ראשית: **יצחק עמיהוד**

בדיקה, ייעוץ VB, עריכה לשונית ועיצוב: **אניקה סואץ**

עיצוב עטיפה: **שרון רז**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי עשתה כמיטב יכולתה למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

Visual Basic הינו מוצר רשום של חברת Microsoft

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך אחריות כלשהי.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי אינה אחראית כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור המצורף.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

- **טלפון: 09-9564716**
- **פקס: 09-9571582**
- **דואר אלקטרוני: Info@hod-ami.co.il**
- **אתר באינטרנט: http://www.hod-ami.co.il**

Visual Basic 6
תכנות משחקי מחשב

ולדיסלב קיפניס, ד"ר
יאנה נחמנסון





הוצאת הוד-עמי
לספרי מחשבים

Visual Basic 6 - Games programming

V. Kipnis Ph.D., Y. Nachmanson

Editor: **I. Amihud**

הוצאת הוד-עמי
לספרי מחשבים בע"מ
ת.ד. 6108 הרצליה 46160
טלפון: 09-9564716 פקס: 09-9571582
דואר אלקטרוני: info@hod-ami.co.il

הודפס בישראל 1999

ISBN 965-361-209-3 מסת"ב

תוכן עניינים מקוצר

תוכן העניינים

פרק 1: התוכנית הראשונה שלי 13

פרק 2: משחק "הזיכרון" 31

פרק 8: משחק "זיכרון 2" 179

פרק 9: משחק "קוד סודי - 2" 193

פרק 10: משחק "שולה מוקשים" 203

נספח: התקליטור המצורף 211

הקדמה

אתם עומדים כעת ליד מדף הספרים בחנות ועליכם להחליט אם לרכוש את הספר הזה עבורכם או עבור מישהו אחר, למשל כמתנה לאחד מבני המשפחה הצעירים. ייתכן גם שאתם פשוט מעוניינים בספר לימוד בסיסי שיאפשר לכם כניסה נעימה לעולם התכנות אך בדרך שונה מזו של הספרים האחרים שעל המדף.

קיימים ספרים רבים ללימוד תכנות ורבים מביניהם הם ספרים ללימוד סביבת הפיתוח של Visual Basic. מדפדוף קל בספר תבין שהוא **אינו ספר רגיל** משום שהוא מיועד ללימוד **תכנות בכיף**. לכאורה, הספר שם דגש על פיתוח משחקים ושעשועוני מחשב, למעשה, תוך כדי הלימוד הוא מציג את יסודות התכנות והפיתוח.

מעט על Visual Basic

Visual Basic הפכה במידה רבה לשפת התכנות הפופולרית. אפשר לכתוב בה תוכניות פשוטות ללא גרפיקה, או מורכבות עם גרפיקה אנימציה וקול ובקצרה – **מולטימדיה**. השאלה היא, כיצד לעשות זאת?

למי מיועד הספר?

הספר מתאים לאנשים היודעים להשתמש ב-Windows ובסייר Windows. לכל הצעירים ברוחם, ולא רק לילדים. אם ברצונכם ללמוד Visual Basic, או שאתם מתלבטים אם תכנות הוא המקצוע שלכם, קבלו את עצתנו : נסו להתחיל בתכנות מספר פרויקטים מתוך הספר וגלו אם יש לכם עניין. אך היזהרו, מרבית הסיכויים שעד מהרה תגלו שככל שאתם מתקדמים, אתם מתמכרים ליופי, לאסתטיקה, לתבונה ולהעשרת הנפש. בהחלט ייתכן שכשתריצו פרויקטים שכתבתם, תבינו שנשביתם בכבלי השפה ובעבותות התכנות, ועם כל פרויקט שמסתיים הנאתכם תגבר, ראו הוזהרתם...

כיצד בנוי הספר?

הספר מבוסס על שיטת הלימוד המיושמת בבית הספר שאנו מנהלים. מעבודתנו בבית הספר למדנו שתוכניות חישוב שכר, ניהול מלאי או יומן רפואי אינן מעניינות את הצעירים, ולמעשה הדבר האהוב עליהם ביותר הם משחקי מחשב; לכן החלטנו ללמדם **לפתח משחקים באופן עצמאי** ולא רק להפעיל משחקים מוכנים. התלמידים הרבים שלמדו בשיטת לימוד זו הפנימו את יסודות הפיתוח, למדו לכתוב תוכניות מחשב פשוטות, ובעיקר - למדו לחשוב באופן עצמאי ולוגי. בוגר שיטת לימוד זו הופך ממשתמש צעיר ממוצע שעיקר עיסוקו הוא משחק במשחקים קנויים, למפתח ומתכנת.

מעבודתנו למדנו גם שצעירים אינם אוהבים לקרוא ספרי לימוד מקובלים ולעקוב אחר ההוראות השונות, משום שאין להם סבלנות, הם רוצים לפעול, לקבל משוב ולעבור לצעד המעשי הבא. לכן, ספר זה ערוך בדרך המתאימה לצעירים. כל פרק הוא פרויקט בפני עצמו, וכל הפרויקטים מסודרים לפי רמת קושי עולה. **פרויקט** אינו סתם דוגמת תכנות; בסיומו יש ללומד **תוכנית מוכנה** שניתן להפעילה ולראות תוצאות! ובסיום העבודה לפי הספר הוא יכיר ויבין את עקרונות הפיתוח ב- Visual Basic. במהלך הלימוד משולבים רוב מושגי התכנות הבסיסיים, ביניהם: הגדרת משתנים, כתיבת פרוצדורות ומודולים, פונקציות ופקודות מובנות, לולאות ועבודה עם מערכים.

על התקליטור

בנספח שבסוף הספר ניתן **הסבר מפורט** על **התקליטור**, תכולתו **והוראות התקנה** לתוכנות שבו. בתקליטור מצורפת גרסת ההפעלה (Working Model) של Visual Basic 6 - תוכנה שבעזרתה אפשר להריץ את התוכניות ללא צורך בתוכנה אחרת. הוצאת הוד-עמי אינה תומכת בשימוש בתוכנה זו. בנוסף, נמצאים בתקליטור קודי התוכניות (פרויקטים) שבספר שנבדקו בגירסה המלאה. המלצתנו, כיתבו את התוכניות לפי ההוראות שבספר, אם תיתקלו בבעיה כלשהי פנו לקודי התוכניות שבתקליטור. שאלות והערות ניתן להפנות למחברים במספר הטלפון בעמוד זה.

תודות מיוחדות

בשני הפרקים האחרונים שבספר בחרנו להציג תוכניות שנכתבו על ידי תלמידי בית הספר: דניאל פפר בן ה-14 ואלקס פרדקין בן ה-15. אנו מודים להם על עבודתם הנאה. אנו מודים גם לילדינו, דניאל נחמנסון בן ה-13 וג׳ניה קיפניס בן ה-12, שעזרו לנו מאוד בעבודה על הספר הזה. לבסוף, תודה גם ליותם סואץ בן ה-14, שבנה את התוכניות ב- Visual Basic 6.

בהצלחה!

ד״ר ולדיסלב קיפניס **יאנה נחמנסון**

טלפון: **02-6232599**

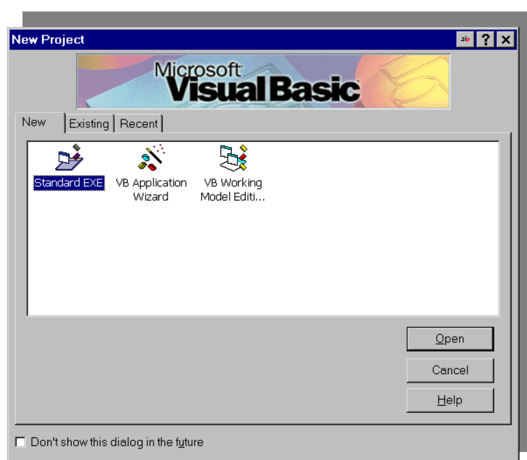
1 התכנית הראשונה שלי

בפרק זה נכיר את השיטות הפשוטות ביותר בתכנות Visual Basic, ונכתוב תוכנית משחק קצרה, שנוכל להתקין אותה במחשב שברשותנו.

הפעלה של Visual Basic 6.0

ספר זה מניח שהתוכנה Visual Basic 6 מותקנת במחשב שלך. אם לא, אנא דאג לרכוש ולהתקין את התוכנה. בתקליטור מצורפת תוכנה שנקראת Visual Basic Working Model שיכולה לעזור לך עד שתקנה את התוכנה. בגירסה המלאה של Visual Basic 6.0 קיימות הרבה יותר אפשרויות, מאשר בגירסה זו. הספר הורץ ונבדק על גרסת התוכנה Visual Basic המלאה ולא על גרסת Working Model. ראה פרטים נוספים בנספח.

לאחר ש- Visual Basic 6.0 תותקן, בסייר Windows נלחץ לחיצה כפולה על הקובץ **VB6.exe**. כאשר ניכנס לחלון העבודה של Visual Basic 6.0, התוכנה תציע לנו לבחור סוג פרויקט. בחלון **New Project** נבחר את סוג הפרויקט **Standard EXE**.



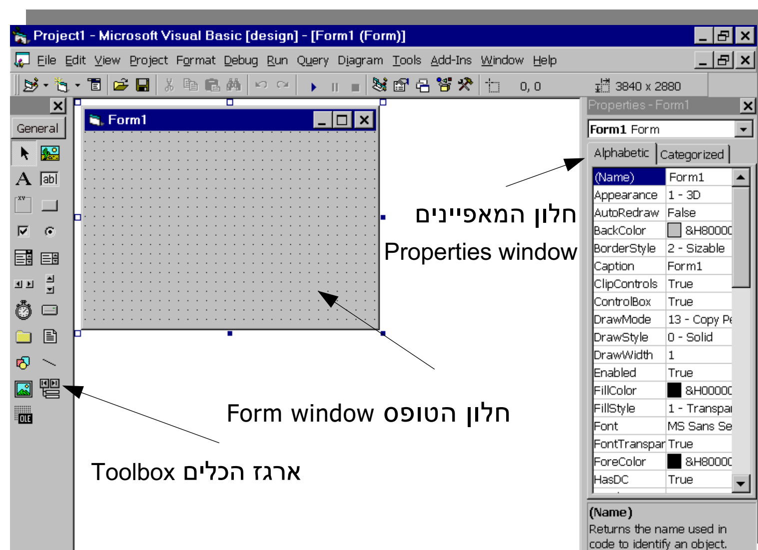
חלון עבודה של Visual Basic 6.0

המרכיב העיקרי של התכנית ב- Visual Basic הוא הטופס (**Form**). טופס זה חלון שבו בונים את ממשק המשתמש ובו התכנית עובדת. בינתיים החלון ריק. דאך גם עכשיו הוא פעיל, ואפשר לבדוק זאת ולראות שהקשה על המקש **F5** מפעילה את התכנית.

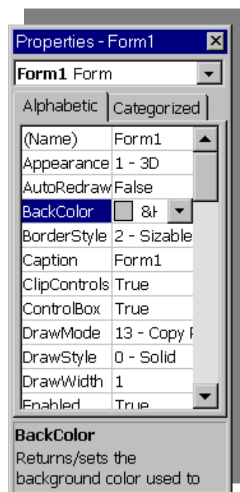
לחלון הטופס יש את כל התכונות של חלון Windows רגיל. אפשר להזיז אותו, לשנות את מידותיו ואת צורתו, וגם להשתמש בלחצנים שכבר קיימים בשורה העליונה שלו.

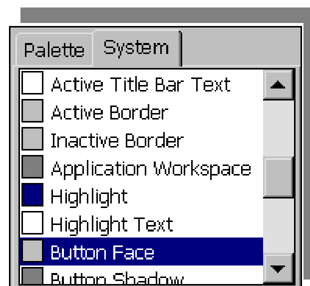
לאורך צידו השמאלי של המסך ממוקם ארגז הכלים (**Toolbox**). ארגז הכלים מכיל מרכיבים סטנדרטיים לתכנות ממשק המשתמש: לחצנים, אובייקטים גרפיים, תוויות, תיבות רשימה וכו'. נכיר אותם במשך עבודתנו.

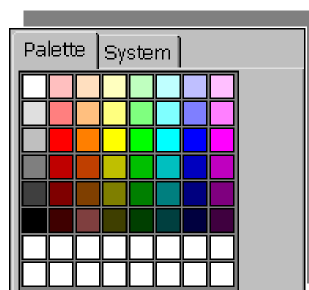
בצד ימין ממוקם חלון המאפיינים (**Properties**). הוא מאפשר לשנות את תכונות מרכיבי ממשק המשתמש. בינתיים המרכיב היחיד של התכנית הינו הטופס - חלון העבודה שלנו, לכן אנו יכולים לשנות את תכונות מרכיב זה. נעשה את זה יחד.



קודם נשנה את צבע הטופס. נשתמש במאפיין שנקרא **Background Color** או בקיצור **BackColor**. אם נלחץ על הלחצן שבשורת המאפיין **BackColor**, ייפתח חלון נוסף - חלון בחירת הצבעים.

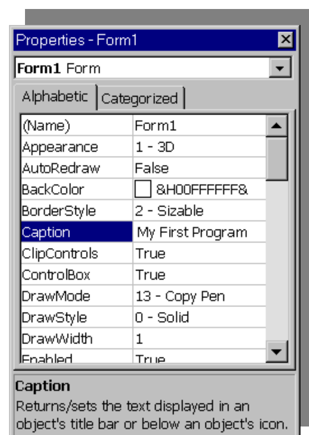


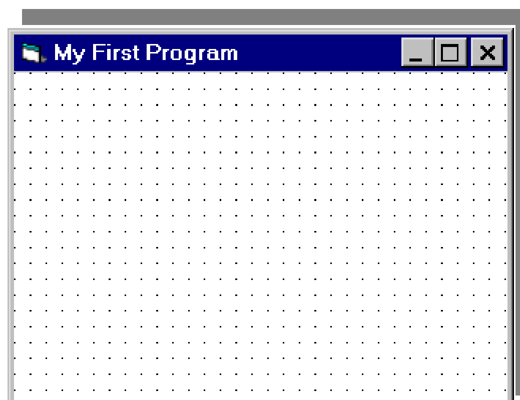




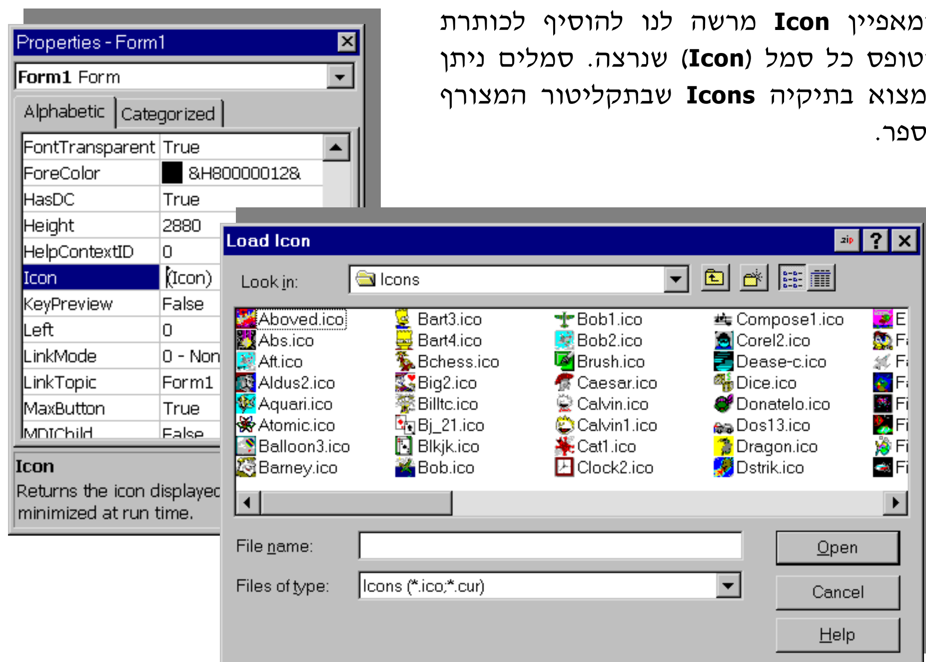
לחלון זה שתי כרטיסיות: **System** ו-**Palette**. נפתח את הכרטיסיה **Palette**, ונראה שאפשר לצבוע את הטופס בכל צבע, אפילו בזה שאינו נמצא במשבצות. צריך רק ללחוץ על אחת מהמשבצות הריקות וייפתח חלון בחירת הצבעים - **Define Color**.

שים לב למאפיין **Caption**, אם תשנה אותו תשתנה הכתובת שבכותרת הטופס.



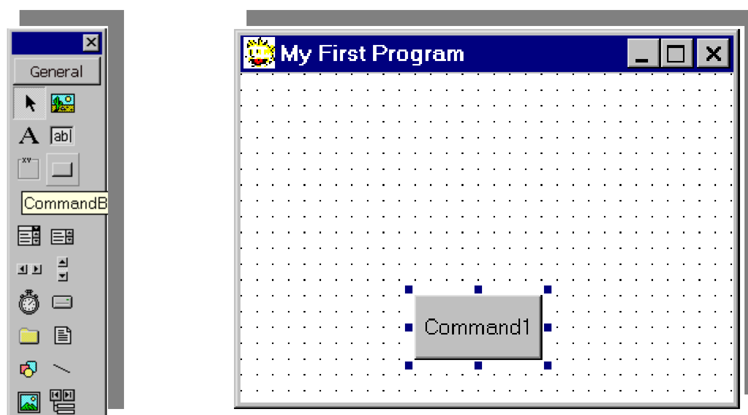


המאפיין **Icon** מרשה לנו להוסיף לכותרת הטופס כל סמל (**Icon**) שנרצה. סמלים ניתן למצוא בתיקיה **Icons** שבתקליטור המצורף לספר.



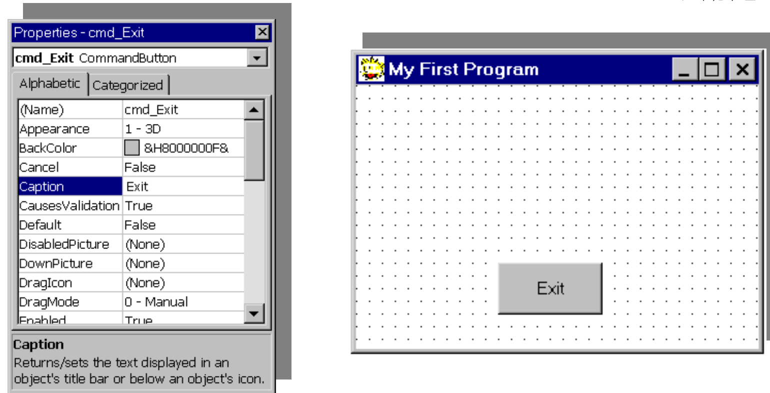
יצירת הלחצן הראשון

1. נבחר לחצן פקודה (**CommandButton**) מארגז הכלים ונוסיף אותו לטופס.



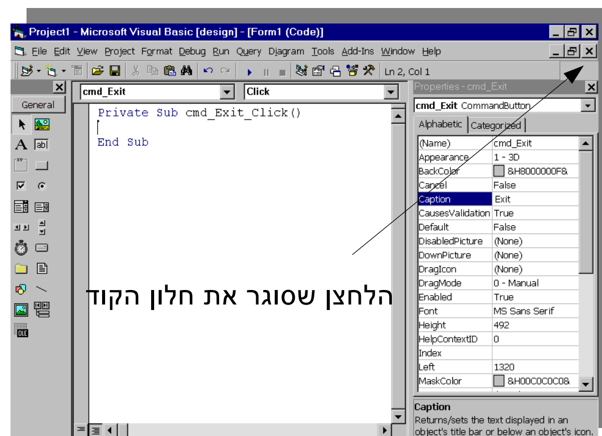
2. נשנה את המאפיין **Caption**: במקום הכיתוב **Command1** ש- Visual Basic מעניקה ללחצן באופן אוטומטי, נכתוב **Exit** (יציאה).

3. ניתן לשנות גם את המאפיין **Name** - שם הלחצן. שמו הקודם היה **Command1** ועכשיו נעניק לו את השם **cmd_Exit**, כלומר, **Command Button Exit** - לחצן פקודת "יציאה".



4. נפעיל עכשיו את התוכנה (**F5**). לחצן **Exit** "מתנהג כמו אמיתי", אבל שום יציאה אינה מתרחשת. למה?

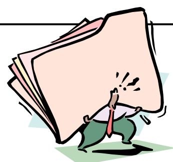
כדי שהלחצן באמת יגרום ליציאה מהתוכנה, צריך לתת לו את הפקודה המתאימה. הפקודה הזאת היא פשוטה וכוללת מילה אחת - **End**. אנו כותבים את הפקודה בחלון מיוחד. אפשר להגיע לשם, אם נלחץ לחיצה כפולה בעכבר על הלחצן **Exit**.



5. ניתן ללחצן היציאה את הפקודה המתאימה - **End**.

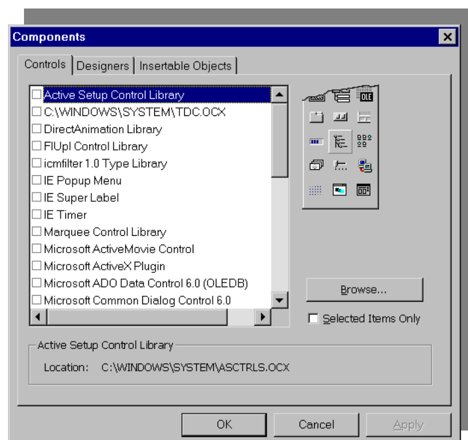
```
Private Sub cmd_Exit_Click()
        End
End Sub
```

6. אנו יכולים להקיש על **F5** ולהשתכנע שהלחצן מבצע את הפקודה.



קצת היסטוריה...

הרבה מרכיבים נוספים "מסתתרים" בתפריט **Project**, בחלק **Components**. אנו לא צריכים אותם בינתיים, אבל ייתכן שעם הזמן נזדקק להם.

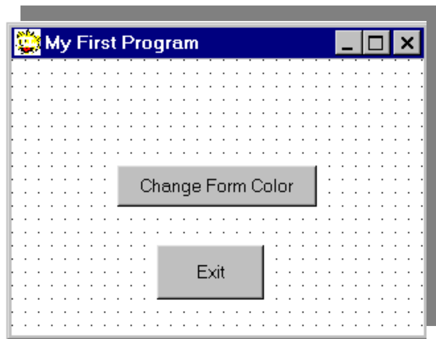


לכל מרכיב של Visual Basic אפשר להתאים מאפיינים מסוימים. מאפיינים אלה יכולים להיות זהים לחלק מהמרכיבים, אך יכולים להיות גם שונים מאוד.

כך למשל, גם למרכיב-טופס וגם ללחצן-פקודה יש מאפיין **BackColor**, אך ללחצן פקודה מאפיין זה יופיע רק לאחר שנשנה את המאפיין **Style** מסטנדרטי - **Standard** לגרפי - **Graphical** (ננסה לשנות את צבע הלחצן **cmd_Exit**). לעומת זאת למרכיב-טופס אין מאפיין **Style**.

שינוי צבע הטופס

1. נוסיף לטופס לחצן פקודה שני ונקרא לו cmd_ChangeColor. נשנה את הכיתוב על הלחצן (מאפיין **Caption**) ל- **Change Form Color**.



2. עכשיו יש להוסיף ללחצן **cmd_ChangeColor** את הפקודה המשנה את צבע הטופס שלנו. למטרה זו נלחץ לחיצה כפולה על הלחצן וניכנס לחלון הקוד.

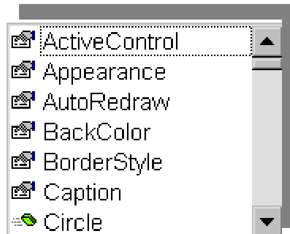
3. כדי לשנות צבע קיים ב-Visual Basic פונקציה בשם **vbColor**, ואת המילה **Color** אפשר להחליף לצבע מסוים, למשל, **vbRed**.

4. נכתוב את הפרוצדורה שתתבצע על ידי הלחצן שלנו:

```
Private Sub cmd_ChangeColor_Click()
        Form1.BackColor = vbRed
End Sub
```

לאחר שנכתוב את הפרוצדורה, נראה כיצד צבע הטופס משתנה, כאשר נלחץ על הלחצן. ובכן, אפשר לשלוט על המאפיינים גם בזמן פעולת התוכנית.

כאשר נכתוב את המילה **Form1** ואחריה נקודה, נפתח לפנינו חלון שבו יכולנו למצוא את המאפיין הנחוץ - **BackColor**.



כדי למצוא מהם את המילה הנחוצה, יש לעבור על המאפיינים וללחוץ פעמיים על אחד מהם.

העבודה תרוץ מהר יותר, אם נתחיל להקליד את האותיות הראשונות של המילה - הסמן יתמקם על המילה הדרושה בלבד.

מי שמתקשה בכתיבת תוכנית זו, יכול לפתוח את התיקיה **VB Projects** שבתקליטור המצורף, למצוא בה את התיקיה **Part 1**, ולהפעיל את הקובץ **MyFirst_1.vbp**.

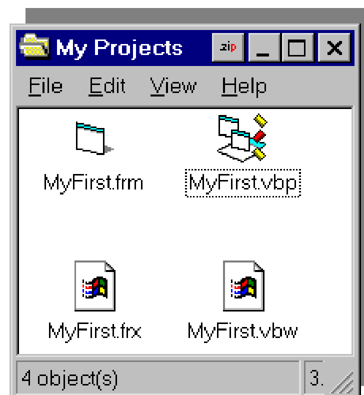
שמירת הקובץ

כדי לשמור את העבודה המוכנה ב- Visual Basic 6.0, נזדקק לשני קבצים. חוץ מזה, Visual Basic יוצרת לפעמים קבצים נוספים.

הקובץ הראשון מיועד לשמירת הטופס, ושני - לשמירת הפרויקט עצמו. כדי לשמור את עבודתנו ניכנס לתפריט **File** ונבחר את הפקודה **Save Project As**.

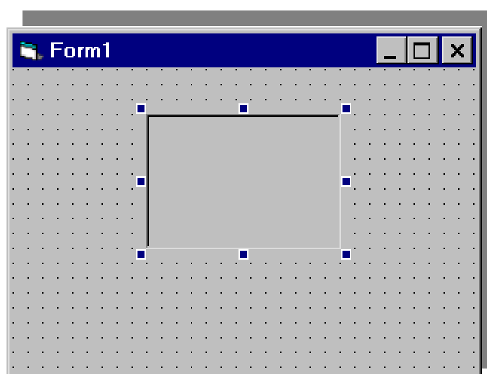
Visual Basic תציע לנו לשמור את קובץ הטופס בשם **Form1.frm**. לא כדאי להסכים להצעה זו: טוב יותר לתת לקובץ הטופס שם מעשי המתאר את שם עבודתנו, למשל, **MyFirst.frm**.

לאחר שמירת הטופס, Visual Basic תפתח מייד את החלון לשמירת קובץ הפרויקט בשם **Project1.vbp**. גם במקרה זה כדאי לתת לקובץ שם מתאים. כדאי ששמות קבצי הטופס והפרויקט יהיו זהים (אך לא סיומות שם הקובץ).



התאמת מאפיינים

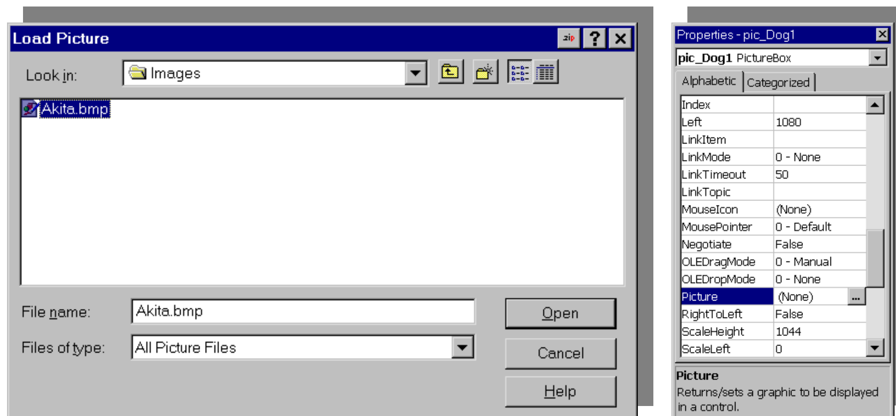
1. נפתח פרויקט חדש ב- Visual Basic - **New Project**.
2. נצייר בחלון הטופס את המסגרת לתמונה - **Picture**.
3. בחלון המאפיינים יש לשנות את שם המסגרת ל-**pic_Dog1**. הקידומת (Prefix) **pic** הינה השם המקוצר של המרכיב **Picture**.
4. נאתר בחלון המאפיינים את המאפיין **Picture**. בעזרתו נכניס את התמונה למסגרת **pic_Dog1**.



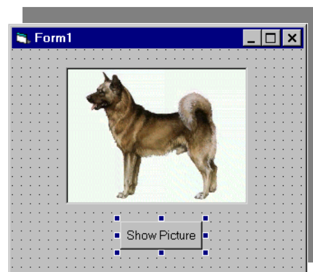
5. נלחץ בעכבר על הלחצן עם שלוש נקודות שבשורת המאפיין **Picture**.

6. נפתח את התיקיה **Images**, אשר נמצאת בתקליטור, ונמצא שם את הקובץ **Akita.bmp**. נוסיף את הקובץ למסגרת **pic_Dog1**.

7. בחלון המאפיינים נאתר את המאפיין **Visible** (גלוי) ונשנה את משמעותו ל-**False**.



8. נוסיף לחצן בשם **cmd_Show** ונשנה את הכיתוב (**Caption**) - **Show Picture** (הצג תמונה).



9. נלחץ על הלחצן לחיצה כפולה וניכנס לחלון הקוד.

10. להלן השיגרה שנכתוב ללחצן:

```
Private Sub cmd_Show_Click()
        pic_Dog1.Visible = True
End Sub
```

11. נפעיל את התוכנית. על המסך יופיע טופס ריק עם הלחצן **Show Picture**. ברגע שנלחץ עליו תופיע תמונת כלב.

12. עכשיו נוסיף לחצן שלישי בשם **cmd_Close**. בשורת המאפיין **Caption** של לחצן זה נכתוב **Hide Picture** (הסתר תמונה).

13. נכתוב את השיגרה:

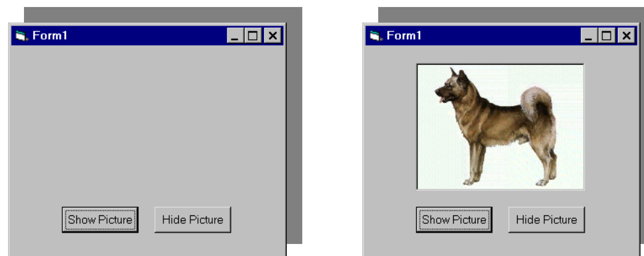
```
Private Sub cmd_Close_Click()
        pic_Dog1.Visible = False
End Sub
```

14. עכשיו התוכנית שלנו מבצעת שתי פעולות: מציגה את התמונה (מאפיין **Visible** שווה **True**) ומסתירה אותה (מאפיין **Visible** שווה **False**).

15. כדי לשמור את הפרויקט יש לפתוח את תפריט **File** ולבחור את הפקודה **Save Project As...**.

16. נשמור את הטופס כקובץ **MyFirst_2.frm**, ולאחר מכן את הפרויקט עצמו - **MyFirst_2.vbp**.

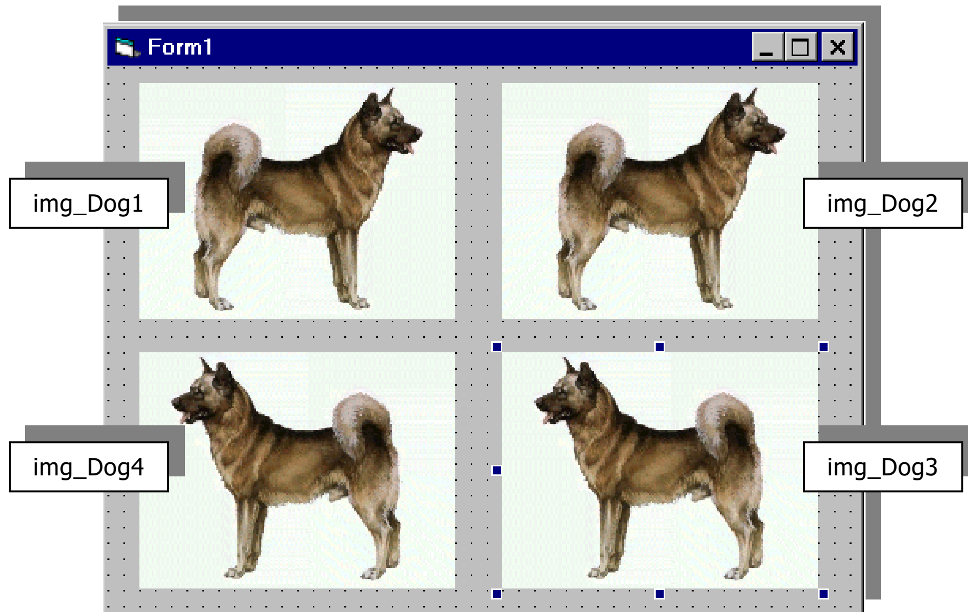
17. נבחר בתפריט את הפקודה **Make MyFirst_2.exe** ונרשום את התוכנית כקובץ ההפעלה (בגרסת Visual Basic 6.0 Working Model הפקודה פסיבית).



כלבים מקפצים

1. נפתח פרויקט חדש: **File - New Project**.
2. נקבע בטופס **Form1** בשורת המאפיין **Width** (רוחב) ערך 5800.
3. למאפיין **Heigth** (גובה) נקבע ערך 4500.
4. בחלק התחתון של ארגז הכלים נמצא את המרכיב **Image**.
5. נצייר על הטופס מסגרת עם מידות כלשהן.
6. נעניק לה שם **img_Dog1** (**img** - הקיצור המקובל ב- Visual Basic למרכיב **Image**).
7. נשתול במסגרת את התמונה-קובץ **Akita2.bmp**.
8. נצייר מסגרת שנייה, נקרא לה **img_Dog2** ונשתול בתוכה את אותו הקובץ **Akita2.bmp**.
9. ניצור עוד שתי מסגרות עם השמות **img_Dog3** ו-**img_Dog4**, ונשתול בתוכן את הקובץ **Akita.bmp**.

10. נסדר את המסגרות, כפי שמצוין בתמונה.



11. נקיש על **Ctrl** ונשאיר אותו לחוץ. במקביל נסמן בעכבר את התמונות האחרות: השנייה, השלישית והרביעית.

12. לכל שלוש המסגרות המסומנות נקבע במאפיין **Visible** ערך **False**.

13. נלחץ לחיצה כפולה בעכבר על המסגרת הראשונה (**img_Dog1**), ניכנס לחלון הקוד ונכתוב את השיגרה לתמונה הראשונה:

```
Private Sub img_Dog1_Click()
        img_Dog1.Visible = False
        img_Dog2.Visible = True
End Sub
```

14. את אותן שגרות נכתוב לשאר התמונות:

```
Private Sub img_Dog2_Click()
        img_Dog2.Visible = False
        img_Dog3.Visible = True
End Sub

Private Sub img_Dog3_Click()
        img_Dog3.Visible = False
        img_Dog4.Visible = True
End Sub

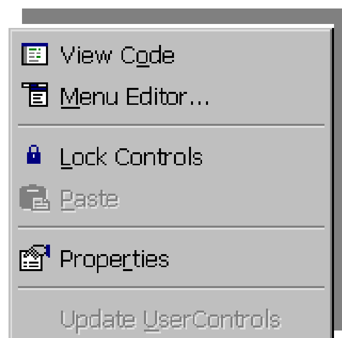
Private Sub img_Dog4_Click()
        img_Dog4.Visible = False
        img_Dog1.Visible = True
End Sub
```

15. כשנפעיל את התוכנית, נראה רק את הכלב הראשון על הטופס. אם נלחץ עליו, הכלב ייעלם והכלב הבא יופיע. לאחר לחיצה נוספת בעכבר ייעלם גם הוא, והכלב השלישי יופיע אחריו.

16. כדי שהרושם של הכלב המקפץ יהיה מציאותי יותר, נקבע לטופס **(Form1)** בשורת המאפיין **BackColor** את הצבע לבן.

כלבים בורחים

אפשר קצת לשנות את התוכנית שלנו. במקום ללחוץ בעכבר על דמות הכלב, מספיק יהיה להציב עליו את הסמן, והכלב "יברח" מאתנו.

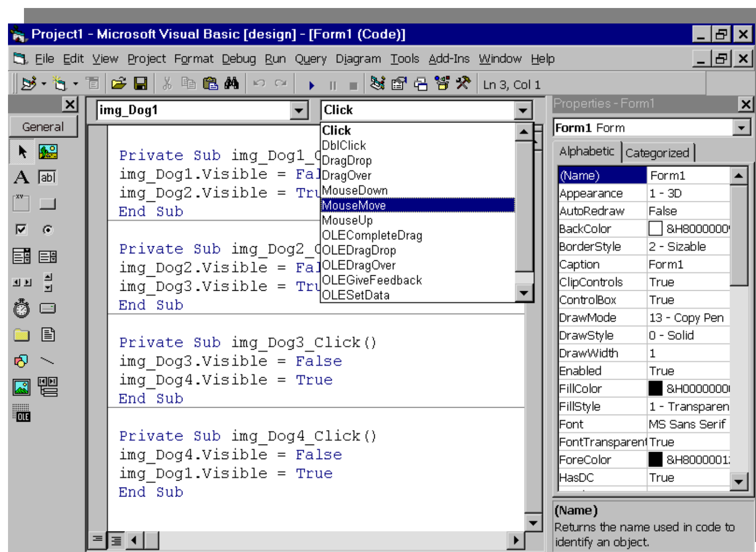


1. כדי לפתוח את חלון הקוד, אפשר ללחוץ על הלחצן הימני של העכבר במקום כלשהו על הטופס, ולקבל את התפריט שבתמונה.

2. בעזרת העכבר נסמן את השורה **View Code** (את אותה פקודה בדיוק ניתן למצוא בתפריט **View** (בשורת התפריטים).

3. נציב את הסמן על השיגרה הראשונה - **Private Sub img_Dog1_Click()**.

4. נפתח את החלון הימני העליון של חלון הקוד ונבחר ברשימה אירוע (**Event**) את **MouseMove**.



5. נעביר את שתי הפקודות מהשגרה **Private Sub img_Dog1_Click()** לשגרה **Private Sub img_Dog1_MouseMove** (כלומר, מעכשיו במקום השגרה לאירוע **Click**, תשמש אותנו השגרה המורשה לאירוע **MouseMove**).

```
Private Sub img_Dog1_MouseMove(Button As Integer, _
Shift As Integer, X As Single, Y As Single)
        img_Dog1.Visible = False
        img_Dog2.Visible = True
End Sub
```

6. נכתוב מחדש באותה דרך את שלוש השגרות ל- **img_Dog2**, **img_Dog3** ול-**img_Dog4** ונפעיל את התוכנית: עכשיו, ברגע שנקרב את הסמן לכלב, הוא יברח מאתנו.

מה עוד יכול לקרות?

במשך עבודתו במחשב למדנו להשתמש בשיטות ניהול שונות: הקשה על לחצני המקלדת, לחיצה רגילה בלחצן העכבר, לחיצה כפולה וכו'. ב- Visual Basic שיטות אלו נקראות אירועים (**Events**).

אירוע הינו אחד מהמושגים הבסיסיים ב- Visual Basic. תוכנית ב-Visual Basic, בנויה כמעט כולה על עיבוד הסימנים הנכנסים - אירועים. למשל, כדי להראות את התמונה, צריך ללחוץ בעכבר על הלחצן **cmd_Show** (אירוע **Click**), כדי לצאת מהתוכנה - יש ללחוץ (**Click**) על הלחצן **Exit**. לכן ב- Visual Basic, השגרות (**Private Sub**) קשורות תמיד לאירועים מסוימים, למשל:

- אירוע - **לחיצה אחת** בעכבר על לחצן הפקודה:

```
Private Sub Command1_Click()
```

- אירוע - **לחיצה אחת** בעכבר על מסגרת תמונה:

```
Private Sub Image1_Click()
```

- אירוע - **לחיצה כפולה** בעכבר על מסגרת תמונה:

```
Private Sub Image1_DblClick()
```

- אירוע - **הזזת העכבר** בגבולות מסגרת התמונה:

```
Private Sub Image1_MouseMove
```

כעת נראה כיצד אפשר להשתמש בסוג נוסף של אירוע:

- **שינוי טקסט** בתיבת טקסט:

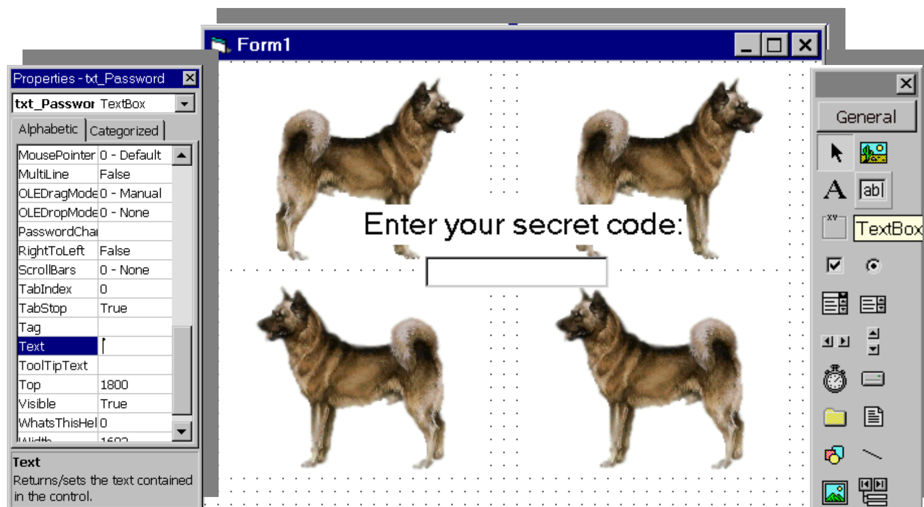
```
Private Sub Text1_Change()
```

הכנת סיסמה

1. נצייר את תיבת הטקסט ישירות על המסגרת עם הכלבים, ונעניק לה שם **txt_Password**.

2. בשורת המאפיין טקסט נמחק את הכיתוב **Text1**.

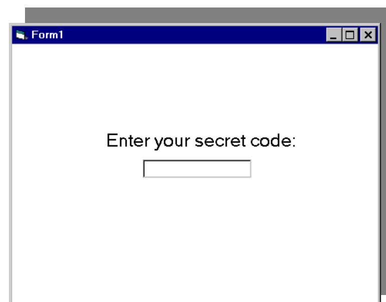
3. על תיבת הטקסט נצייר תווית **Label**, שממוקמת באראגז הכלים ליד תיבת הטקסט. במאפיין **Caption** של התווית אפשר לכתוב: **Enter your password** או: **Enter your secret code**.



4. נצבע את התווית בצבע לבן (מאפיין **BackColor**), נציב את הכיתוב במרכז התווית (מאפיין **Center - Alignment**), נבחר גודל גופן מתאים (מאפיין **Font**) ונצבע את הכיתוב (מאפיין **ForeColor**). אם כתוצאה מהגדלת הגופן הכיתוב לא נכנס לתווית, יש לגרור אותה באמצעות העכבר.

5. את מסגרת הכלב הראשון **img_Dog1** נהפוך לבלתי-נראית (מאפיין **Visible - False**).

6. כעת כשנפעיל את התוכנית, יופיע על המסך טופס עם תיבת הטקסט וכתובת.



7. בתיבת הטקסט נכתוב את השיגרה לשינוי הטקסט:

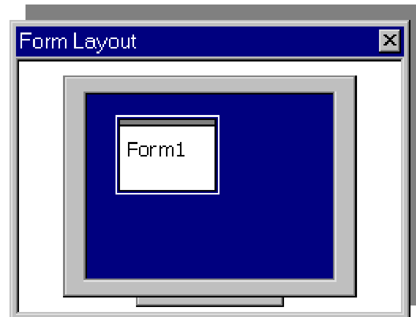
```
Private Sub txt_Password_Change()
If txt_Password.Text = "Daniel" Then
            txt_Password.Visible = False
            Label1.Visible = False
            img_Dog1.Visible = True
End If
End Sub
```

8. נפעיל את התוכנית. כשתופיע השאלה **Enter your secret code** נקליד את הסיסמה - "Y" (אפשר להשתמש בכל סדרת סמלים, אך יש לרשום אותה בתנאי: **If txt_Password.Text = "..." Then** במקום שלוש הנקודות). כעת, לאחר הקלדת הסיסמה הנכונה, התוכנית תסיר את תיבת הטקסט (**txt_Password.Visible = False**) ואת הכיתוב (**Label1.Visible = False**), ובמקומם תופיע המסגרת עם הכלב הראשון (**img_Dog1.Visible = True**).

עיצוב הטופס הסופי

כדי שהתוכנה תהיה יפה, יש לשנות מעט את הטופס.

1. את מיקום הטופס על המסך אפשר לקבוע בחלון עיצוב הטופס, שנמצא בתפריט **View** שבשורת התפריטים.
2. בחלון עיצוב הטופס נזיז את הטופס ונציב אותו במקום הנכון במסך.
3. בחלק העליון של הטופס בשורת הכותרת (**Title Bar**) נשנה את הכותרת **Form1** לכותרת אחרת, למשל **Jumping dogs**. את השינויים האלה נבצע בשורת המאפיין **Caption** של **Form1**. אפשר לכתוב גם בעברית, אך אז יש לקבוע שהמאפיין **RightToLeft** שווה **True**.



4. בשורת המאפיין **Icon** אפשר לבחור את הסמל המתאים להתקנה בפינה השמאלית העליונה של שורת הכותרת.



5. איננו רוצים שגודל הטופס שלנו ישתנה, לכן נשנה את המאפיין **BorderStyle** מ-**Sizable** ל-**Fixed Single**, כדי שהמשתמש לא יוכל להגדיל את חלון הטופס עד לגודל מסך המחשב.

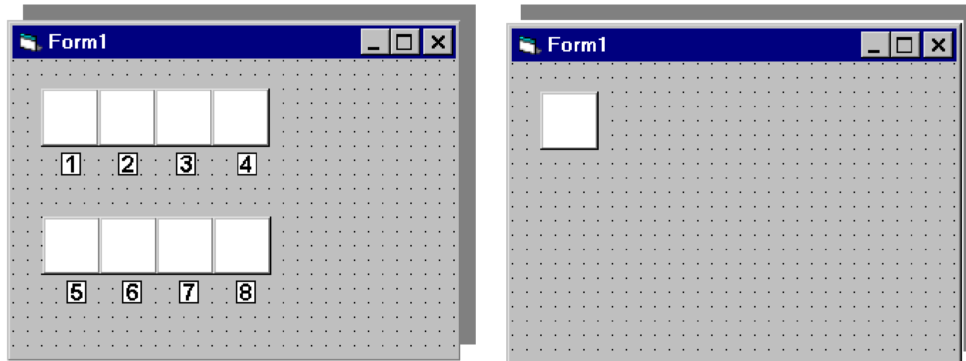
2 משחק "הזיכרון"

קיימות גרסאות מחשב רבות של משחק פופולרי זה. בפרק זה נכתוב תוכנית של הגרסה הפשוטה ביותר - משחק עם 8 תמונות בלבד. קל מאוד לשחק בו: צריך לפתוח אחת משמונה התמונות, לזכור את מקומה, ולאחר מכן לפתוח תמונה זהה לה. אם התמונות הפתוחות זהות - הן יוצאות מהמשחק, ויש לחפש זוגות בין אלו שנשארו. אם התמונה השנייה שונה מהראשונה, שתיהן נסגרות.

כמובן, משחק זה רחוק מהמשחק האמיתי, הרי לזכור מיקום של שמונה תמונות אינו קשה במיוחד. אך בואו לא נמהר להגיע לסיכומים! בתוכנית יש משהו, שעושה את המשחק שלנו לא עד כדי כך קל...

יצירת טופס המשחק

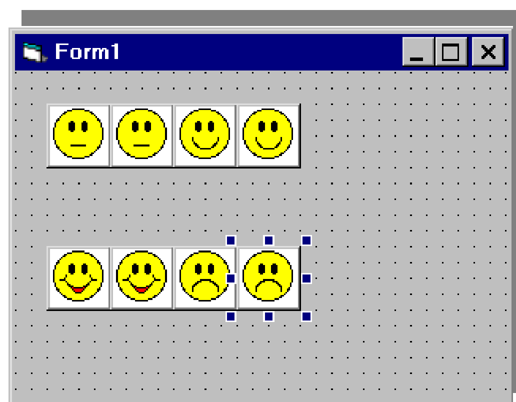
1. נפתח פרויקט חדש (**New Project**) ב- Visual Basic.
2. נוסיף לטופס לחצן פקודה (**CommandButton**) ונקרא לו (מאפיין **Name**) - **cmd_Face1**.
3. נקבע את מידותיו: גובה (מאפיין **Height**) = **492** ורוחב (מאפיין **Width**) = **492**.
4. נשנה את המאפיין **Style** מ-**Standard** ל-**Graphical**.
5. בשורת המאפיין **Caption** נמחק את הכיתוב **Command1**.
6. נצבע את הלחצן בצבע לבן (מאפיין **BackColor**).



7. נעתיק את הלחצן (Copy), נבחר את הפקודה Paste, ועל השאלה: **"You already have a control named 'cmd_Face1'. Do you want to create a control array?"** נענה - **No**.

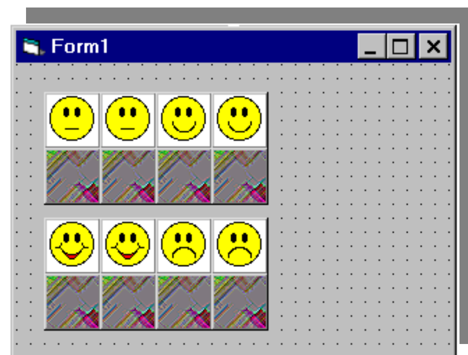
8. נעניק ללחצן את השם **cmd_Face2**. נחזור על פעולה זו עד שנייצור 8 לחצנים כאלה עם השמות מ-**cmd_Face2** עד **cmd_Face8**. נמקם אותם על הטופס בסדר שצוין בתמונה.

9. נשתמש במאפיין **Picture** לכל אחד מהלחצנים, כדי לשתול בהם את התמונות (**Face01.ico** - **Face04.ico**), הנמצאות בתיקיה **Icons** שבתקליטור. לכל זוג לחצנים יהיה זוג תמונות זהות, כפי שצוין בתמונה.



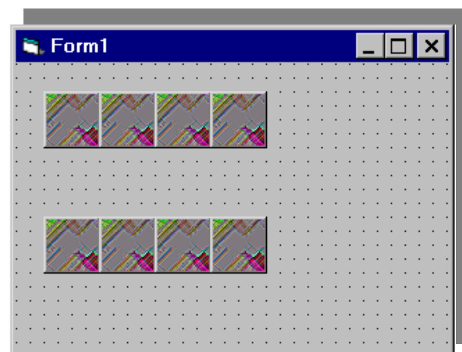
10. נסמן בעכבר את כל הלחצנים ונעניק להם את המאפיין **Visible - False**.

11. נוסיף על הטופס עוד שמונה לחצנים כאלה עם השמות מ-**cmd_Card1** עד **cmd_Card8**. הם ישמשו כ"מכסים" לתמונות שלנו.

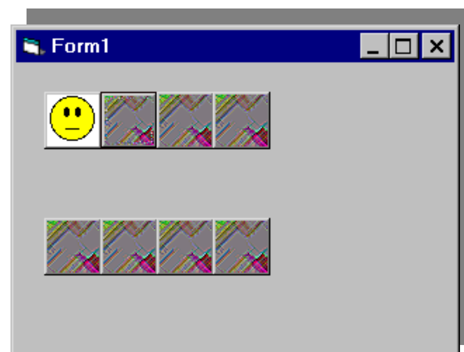


12. נסדר אותם לפי מספרי הלחצנים **cmd_Face1 ...cmd_Face8** ונשתול לכל הלחצנים **cmd_Card1 ...cmd_Card8** תמונות זהות (קובץ **Card.bmp** מהתיקיה **Images**).

13. נכסה כל תמונה ב"מכסה" שלה. מעכשיו התמונות צריכות להיפתח עם לחיצה על הלחצן-"מכסה".



14. נפעיל את התוכנית: נלחץ לחיצה בעכבר על "המכסה" הראשון, הוא "יעלם" ובמקומו תופיע התמונה הראשונה.



15. להלן הפרוצדורות הזהות לשאר הלחצנים:

```
Private Sub cmd_Card2_Click()
        cmd_Card2.Visible = False
        cmd_Face2.Visible = True
End Sub

Private Sub cmd_Card3_Click()
        cmd_Card3.Visible = False
        cmd_Face3.Visible = True
End Sub

Private Sub cmd_Card4_Click()
        cmd_Card4.Visible = False
        cmd_Face4.Visible = True
End Sub

Private Sub cmd_Card5_Click()
        cmd_Card5.Visible = False
        cmd_Face5.Visible = True
End Sub

Private Sub cmd_Card6_Click()
        cmd_Card6.Visible = False
        cmd_Face6.Visible = True
End Sub

Private Sub cmd_Card7_Click()
        cmd_Card7.Visible = False
        cmd_Face7.Visible = True
End Sub

Private Sub cmd_Card8_Click()
        cmd_Card8.Visible = False
        cmd_Face8.Visible = True
 End Sub
```

מתכנתים את מה שעלול לקרות ואת מה שלא

אם נפתח את כל הלחצנים-"מכסים" אחד אחרי השני, תיפתחנה בסופו של דבר כל התמונות ולא יישאר על מה ללחוץ. אך במשחק אמיתי הכל מתרחש אחרת, כאשר אנו פותחים את התמונה הלא נכונה, התמונה שהיתה פתוחה לפני כן נסגרת. לכן יש לצפות גם לאפשרות כזו, כלומר, לכתוב בתוכנית על משהו שיכול לקרות, או לא.

שחקן לא חייב לפתוח את תמונה מספר 1 דווקא. אם נגיד שהתמונה הראשונה שנפתחה היתה תמונה מספר 3, ולאחר מכן נפתח את תמונה מספר 1, תמונה מספר 3 חייבת להיסגר. אותו דבר בדיוק יקרה אם תמונות מספר 2, 5, 6, 7 או 8 היו פתוחות קודם לכן, רק אם תמונה מספר 4 היתה הפתוחה, שתיהן, 3 ו-4 תצאנה מהמשחק.

מכיון שכל התמונות שלנו מסודרות בזוגות: 1 ו-2, 3 ו-4, 5 ו-6, 7 ו-8, השגרות לתמונות זוגיות זהות.

נכתוב קודם את השיגרה ללחצן 1, ולאחר מכן נעביר אותה לשאר הלחצנים.

```
Private Sub cmd_Card1_Click()
        cmd_Card1.Visible = False
        cmd_Face1.Visible = True

        cmd_Face3.Visible = False          ' מסתירים את תמונה מס' 3
        cmd_Face4.Visible = False          ' מסתירים את תמונה מס' 4
        cmd_Face5.Visible = False          ' מסתירים את תמונה מס' 5
        cmd_Face6.Visible = False          ' מסתירים את תמונה מס' 6
        cmd_Face7.Visible = False          ' מסתירים את תמונה מס' 7
        cmd_Face8.Visible = False          ' מסתירים את תמונה מס' 8

        cmd_Card3.Visible = True           ' חושפים את "מכסה" מס' 3
        cmd_Card4.Visible = True           ' חושפים את "מכסה" מס' 4
        cmd_Card5.Visible = True           ' חושפים את "מכסה" מס' 5
        cmd_Card6.Visible = True           ' חושפים את "מכסה" מס' 6
        cmd_Card7.Visible = True           ' חושפים את "מכסה" מס' 7
        cmd_Card8.Visible = True           ' חושפים את "מכסה" מס' 8
End Sub
```

בשיגרה זו כתבנו פקודות "לכל מקרה". אירועים המתוכנתים בשיגרה יכולים לקרות, ויכולים שלא. למשל, תמונה מספר 6 לא חייבת להיפתח, ואחריה לא חייבת להיפתח תמונה מספר 1. אך אם לא נביא בחשבון מקרה כזה, שגיאה זו תתגלה בזמן כלשהו.

כשכותבים את התוכנית, מועיל להוסיף הסברים (הערות - **Comments**). אפשר לכתוב אותם בחלון הקוד, ולהבדיל בינם לבין הקוד באמצעות סימן "'". Visual Basic "יודעת" שכל מה שכתוב לאחר הסימן לא שייך לקוד, ואינה קוראת זאת. ההסברים עוזרים לעקוב אחרי דרך התכנות ולהגיע לשורש העניין בתוכניות מורכבות ומסובכות, במיוחד כשעבר זמן מסוים והמתכנת עצמו כבר אינו זוכר מה כתב בתוכנית.

איננו צריכים לכתוב את כל הפקודות מחדש. מספיק לסמן את הקטע מהשיגרה **Private Sub cmd_Card1_Click()** ולהעתיק אותו (פקודות **Paste - Copy**) לטקסט השגרות בכל שאר הלחצנים (**cmd_Card8...cmd_Card2**), ולשנות את מספרי הלחצנים:

```
Private Sub cmd_Card2_Click()
        cmd_Card2.Visible = False
        cmd_Face2.Visible = True

        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub

Private Sub cmd_Card3_Click()
        cmd_Card3.Visible = False
        cmd_Face3.Visible = True

        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card4_Click()
        cmd_Card4.Visible = False
        cmd_Face4.Visible = True

        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card5_Click()
        cmd_Card5.Visible = False
        cmd_Face5.Visible = True

        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card6_Click()
        cmd_Card6.Visible = False
        cmd_Face6.Visible = True

        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card7_Click()
        cmd_Card7.Visible = False
        cmd_Face7.Visible = True

        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
End Sub
```

```
Private Sub cmd_Card8_Click()
	cmd_Card8.Visible = False
	cmd_Face8.Visible = True

	cmd_Face1.Visible = False
	cmd_Face2.Visible = False
	cmd_Face3.Visible = False
	cmd_Face4.Visible = False
	cmd_Face5.Visible = False
	cmd_Face6.Visible = False

	cmd_Card1.Visible = True
	cmd_Card2.Visible = True
	cmd_Card3.Visible = True
	cmd_Card4.Visible = True
	cmd_Card5.Visible = True
	cmd_Card6.Visible = True
End Sub
```

עכשיו ננסה לשחק

נפעיל את התוכנית וננסה לשחק. אם כל הפקודות כתובות נכון, הלחצנים שעל המסך צריכים לפתוח ולסגור את התמונות אחת אחרי השנייה, חוץ מאלו המהוות זוג.

אך לא הכל מושלם בתוכניתנו. שתי התמונות הפתוחות אינן נשארות על המסך הרבה זמן, לחיצה על הלחצן הבא, גורמת לכיסוי שתי התמונות האלו על ידי ה"מכסים" שלהן, והתחלת המשחק מחדש. לכן צריך לגרום לכך שכאשר שתי תמונות זהות פתוחות, זוג זה ייצא מהמשחק. למשל, אפשר להגדיר במילים הבאות את התנאי שצריך להתבצע על תמונות 1 ו-2:

אם (התמונה הראשונה פתוחה) **ו-** (התמונה השנייה פתוחה) **אז...**

נתרגם את זה לשפת Visual Basic:

```
If (cmd_Face1.Visible = True) And (cmd_Face2.Visible = True) Then
```

אם תנאי זה מתמלא, נגרום לכך ששתי התמונות יחד עם "המכסים" שלהן יפסיקו לתפקד, כלומר, נקבע שמאפיין **Enabled** שלהם שווה **False**:

```
If(cmd_Face1.Visible = True) And _
  (cmd_Face2.Visible = True) Then
      cmd_Face1.Enabled = False            ' "מכבים" את תמונה מס' 1
       cmd_Face2.Enabled = False           ' "מכבים" את תמונה מס' 2
       cmd_Card1.Enabled = False           ' "מכבים" את "מכסה" מס' 1
       cmd_Card2.Enabled = False           ' "מכבים" את "מכסה" מס' 2
End If                                     ' סוף התנאי IF
```

להלן הטקסט המלא של השיגרה ללחצן **cmd_Card1**:

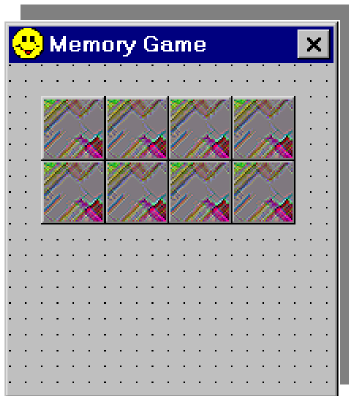
```
Private Sub cmd_Card1_Click()
        cmd_Card1.Visible = False
        cmd_Face1.Visible = True
                    If (cmd_Face1.Visible = True) And _
                       (cmd_Face2.Visible = True) Then
                              cmd_Face1.Enabled = False
                              cmd_Face2.Enabled = False
                              cmd_Card1.Enabled = False
                              cmd_Card2.Enabled = False
                    End If
          cmd_Face3.Visible = False
          cmd_Face4.Visible = False
          cmd_Face5.Visible = False
          cmd_Face6.Visible = False
          cmd_Face7.Visible = False
          cmd_Face8.Visible = False

          cmd_Card3.Visible = True
          cmd_Card4.Visible = True
          cmd_Card5.Visible = True
          cmd_Card6.Visible = True
          cmd_Card7.Visible = True
          cmd_Card8.Visible = True
End Sub
```

כעת נותר רק להעתיק טקסט זה לשגרות הלחצנים האחרים, ולשנות את מספרי המרכיבים, שאותם צריך להוציא מהמשחק.

ראינו שכל שיגרה הינה בעלת מראה מדורג. זה נוח מאוד, משום שאפשר לראות מייד את תחילת התנאי **If** ואת סופו - **End If**.

המשך עיצוב הטופס



1. כעת צריך להחליף את מיקום הלחצנים על הטופס, כדי שלחצנים זוגיים לא יימצאו אחד ליד השני. יחד עם זה, צריך להקפיד שכל תמונה תישאר עם "המכסה" שלה, כלומר, צריך להזיז אותם יחד - את התמונה ואת "המכסה"!
2. במאפיין **Caption** של שורת הכותרת של הטופס שלנו נכתוב "**Memory game**" ונשנה את הסמל הסטנדרטי לאחד מהסמלים שבתיקיה **Icons** בתקליטור.
3. נקבע את המאפיין **BorderStyle** כ-**Fixed Single**.

איך להיפטר מהפוקוס

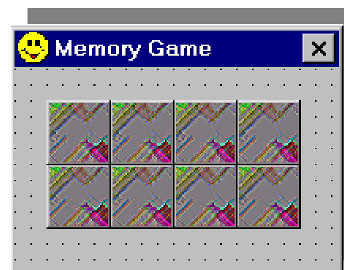
אם נפעיל את התוכנית ונקיש **Enter**, התמונות תיפתחנה מייד, אחת אחרי השנייה בלי שום קשר למקום שלהן.

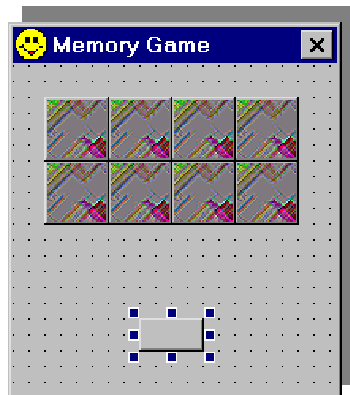
נחזור על פעולה זו וננסה להבין מה קורה עם הלחצנים. הפעם נעשה את הניסוי בעזרת העכבר, כדי לשים לב לשינויים.

1. אנו רואים שאחד מהלחצנים נבדל מהאחרים. נלחץ בעכבר על לחצן זה.
2. עכשיו, במקום הלחצן הראשון הופיע עוד לחצן הנבדל מהאחרים. אם נלחץ בעכבר על לחצן זה, "הפוקוס" יעבור ללחצן הבא. איך להיפטר ממנו?
3. קודם כל "פוקוס" הינו המסגרת הפעילה, המחכה לפקודה - תופעה רגילה ב-Windows. אם נלחץ על לחצן זה באמצעות העכבר או נקיש **Enter**, נקבל את אותה תוצאה. זאת הסיבה שהצלחנו לפתוח את כל התמונות מייד!
4. כדי להיפטר מ"הפוקוס" נוסיף לחלק התחתון של הטופס לחצן קטן ונעניק לו שם **cmd_Focus**.
5. בתחילת כל שיגרה של לחצני **cmd_Card1 ...cmd_Card8** נכתוב את הפקודה:

```
cmd_Focus.SetFocus
```

6. נזיז למעלה את המסגרת התחתונה של הטופס, כדי שתסתתר את הלחצן **cmd_Focus**. מעכשיו לחצן זה לא ייראה בזמן המשחק, אך בכל פעם שנלחץ על אחד מהלחצנים **cmd_Card1 ...cmd_Card8** הפוקוס יעבור אליו.





נשכלל את המשחק

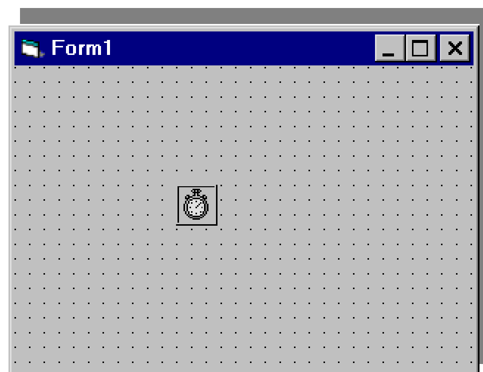
נהפוך את המשחק שלנו ליותר מורכב ומעניין. למשל, אם שחקן לא הספיק לפתוח את התמונות ב-15 שניות, הן תיעלמנה - המשחק ייגמר!

ב- Visual Basic יש כלי מיוחד - טיימר, שבעזרתו אפשר לקבוע מרווחי זמן שונים. טיימר דומה מאוד לשעון עצר רגיל, אך ניתן לשנות את מהירות "התקתוקים" שלו. אפשר "לשמוע" את הטיימר, כפי ששומעים שעון.

כדי להבין יותר טוב איך טיימר בנוי, נכתוב תוכנית קטנה.

1. נפתח פרויקט חדש ב- Visual Basic : **File -Project**.

2. נוסיף לטופס שלנו טיימר. גודל המסגרת שלו נשאר קבוע תמיד.



3. בלחיצה כפולה על הטיימר, ניכנס לחלון הקוד ונכתוב את השיגרה הפשוטה:

```
Private Sub Timer1_Timer()
        Beep
 End Sub
```

4. נפתח את חלון המאפיינים שלו ונקבע את הערך למאפיין **Interval -1000**.

5. נפעיל את התוכנית: מרמקול המחשב נשמע קול מקוטע - **Beep**. כך עובד הרמקול. כל שנייה הטיימר נדלק ונכבה: הוא מבצע את הפקודה **Beep**.

בפרק 1 דיברנו כבר על אירועים, שאפשר לנהל אותם באמצעות מרכיבי פקד של Visual Basic. אחד מאירועים אלה הוא אירוע **Timer**. הוא מזכיר לנו תקתוק שעון רגיל - "תיק-תק-תיק". אפשר לשנות את המרווח בין "התיקים" ואז "השעון" שלנו יתקתק מהר או לאט יותר. הערך **Interval - 1000** שווה בדיוק לשנייה אחת, או 1000 אלפיות השנייה.

Properties - Timer1
Timer1 Timer
Alphabetic | Categorized

(Name)	Timer1
Enabled	True
Index	
Interval	1000
Left	120
Tag	
Top	960

Interval
Returns/sets the number of milliseconds between calls

ועכשיו אנו יכולים לחזור למשחק שלנו.

6. נקבע בטופס שלנו טיימר ונגדיר בשבילו את הערך **Interval - 1000** ומאפיין **False - Enabled** (זאת אומרת שהטיימר שלנו נכבה בינתיים).

7. נוסיף לטופס לחצן בשם (מאפיין **Name**) **cmd_Start** וכיתוב (מאפיין **Caption**) - **Start**. ללחצן זה צריך לכתוב פקודה אחת בלבד - הדלקת הטיימר.

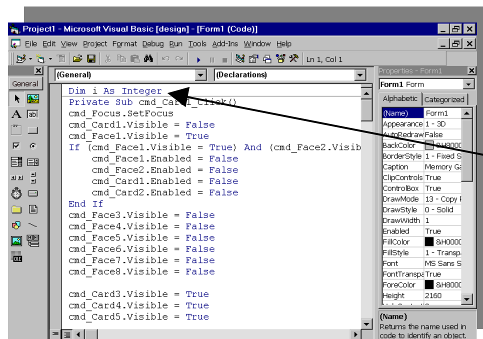
```
Private Sub cmd_Start_Click()
        Timer1.Enabled = True
End Sub
```

איך נוכל לספור את "התקתוקים" של הטיימר? למטרה זו נצטרך מונה - תא מיוחד בזיכרון המחשב. עם כל "תקתוק" חדש נוסף 1 למונה. כשבמונה יצטבר מספר מסוים, הטיימר יתנתק, והלחצנים עם התמונות ייעלמו.

1. נפתח את חלון הקוד ונמקם את הסמן בחלק העליון של החלון, בתחילת התוכנית שלנו. ניתן למונה שם:

```
Dim i As Integer
```

בתכנות זה נקרא ***הגדרת משתנה***. המונה שלנו הינו משתנה בשם **i** מסוג **Integer**, כלומר הערכים של המשתנה הזה יכולים להיות מספרים שלמים בלבד, ולא מספרים עם שבר. מעכשיו התוכנית יודעת שבזיכרון המחשב יש תא בשם **i**, שיספור את "התקתוקים" של הטיימר.



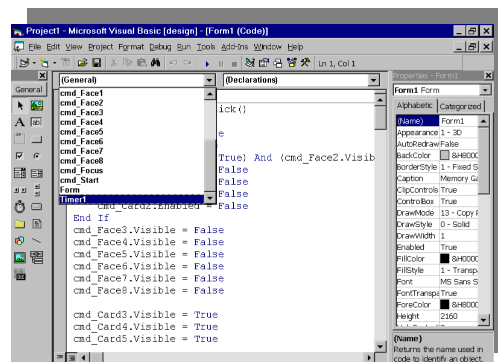
חלק התוכנית **General** שבו אנו מגדירים משתנים בדרך כלל.

2. בחלון קוד נפתח את החלון עם רשימת המרכיבים ונבחר מרכיב בשם **Timer1**.

3. נכתוב את השגרה של הטיימר:

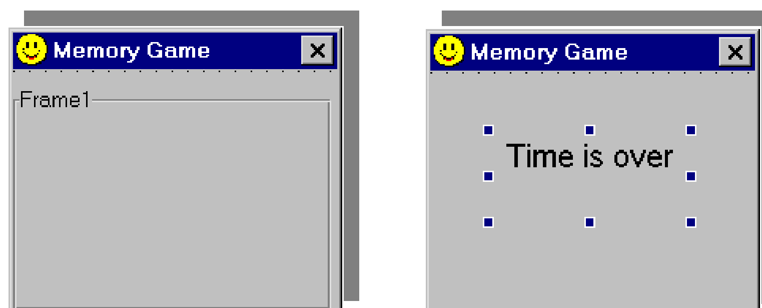
```
Private Sub Timer1_Timer()
       i = i + 1
       Form1.Caption = "Your time is " & i
End Sub
```

4. נפעיל את התוכנית ונלחץ בעכבר על הלחצן **Start**, בשורת הכותרת יופיע



הכיתוב: **Your time is 1** וחישוב "התקתוקים" ימשך. הטיימר שלנו עובד!

5. נפתח את חלון עיצוב הטופס ונצייר מעל כל הלחצנים מרכיב-מסגרת **Frame**.



6. בחלון המאפיינים נמחק את הכיתוב (**Caption**) **Frame1**, למאפיין **BorderStyle** נקבע ערך **0** - **None** ולמאפיין **Visible** - ערך **False**.

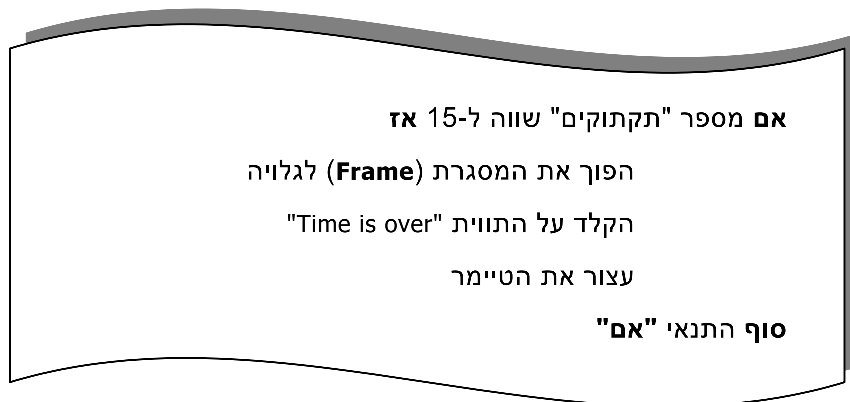
7. נמקם על המסגרת (**Frame**) את התווית (**Label**). בחלון המאפיינים בשורת **Caption** נקליד כיתוב - **Time is over**.

8. נפתח את מאפיין **Alignment** ונקבע בשבילו ערך **2-Center**. מעכשיו הכיתוב על התווית יהיה במרכז.

9. נבחר לכיתוב שלנו את גודל הגופן המתאים (מאפיין **Font**) וצבע האותיות (מאפיין **ForeColor**).

10. ועכשיו נמחק את הכיתוב **Time is over** שבשורת **Caption**.

11. נפתח שוב את חלון הקוד ונכתוב בשגרת הטיימר את התנאי הבא:



בשפת Visual Basic זה נראה כך:

```
If i = 15 Then
        Frame1.Visible = True
        Label1.Caption = "Time is over"
        Timer1.Enabled = False
 End If
```

12. נפעיל את התוכנית וננסה לשחק: אם במשך 15 שניות לא נצליח לגמור את המשחק, המסגרת (**Frame**) תסגור את כל הלחצנים ויופיע כיתוב שהזמן נגמר. ואם הצלחנו לגמור בזמן?

כשכל התמונות פתוחות...

לפי כללי התוכנית שלנו, כל זוג תמונות פתוחות יוצא מהמשחק, זאת אומרת, מאפיין **Enabled** מקבל משמעות **False**. את התנאי אשר על פיו כל התמונות הופכות ל"מתות" אפשר לכתוב כך:

```
If  cmd_Face1.Enabled = False And _
    cmd_Face2.Enabled = False And _
    cmd_Face3.Enabled = False And _
    cmd_Face4.Enabled = False And _
    cmd_Face5.Enabled = False And _
    cmd_Face6.Enabled = False And _
    cmd_Face7.Enabled = False  And _
    cmd_Face8.Enabled = False  Then
```

ואחר כך להגדיר את מה שצריך להתרחש במקרה זה:

```
    Frame1.Visible = True
    Label1.Caption = "Your time is " & i
    Timer1.Enabled = False
```

כלומר, תופיע המסגרת (**Frame1**) עם התווית (**Label1**) שכתוב בה **Your time is**. הזמן המדויק שבו הצלחנו לסיים את המשחק מוגדר על ידי הביטוי **& i**. הביטוי הזה נמצא במירכאות, מוסכמה שמשמשת ב-Visual Basic כהוראה לפנות לתא הזיכרון בשם **i** ולקחת ממנה מספר מוגדר - מספר ה"תקתוקים" של הטיימר. לאחר מכן הטיימר (**Timer1**) צריך להיות מכובה. כדי שבתחילת המשחק הספירה תתחיל מ-0 ולא מ-1, יש להעביר את הפקודה להגדלת מונה ה"תקתוקים" (**i = i + 1**) לסוף ההליך. טקסט שגרת הטיימר השלם ייראה עכשיו כך:

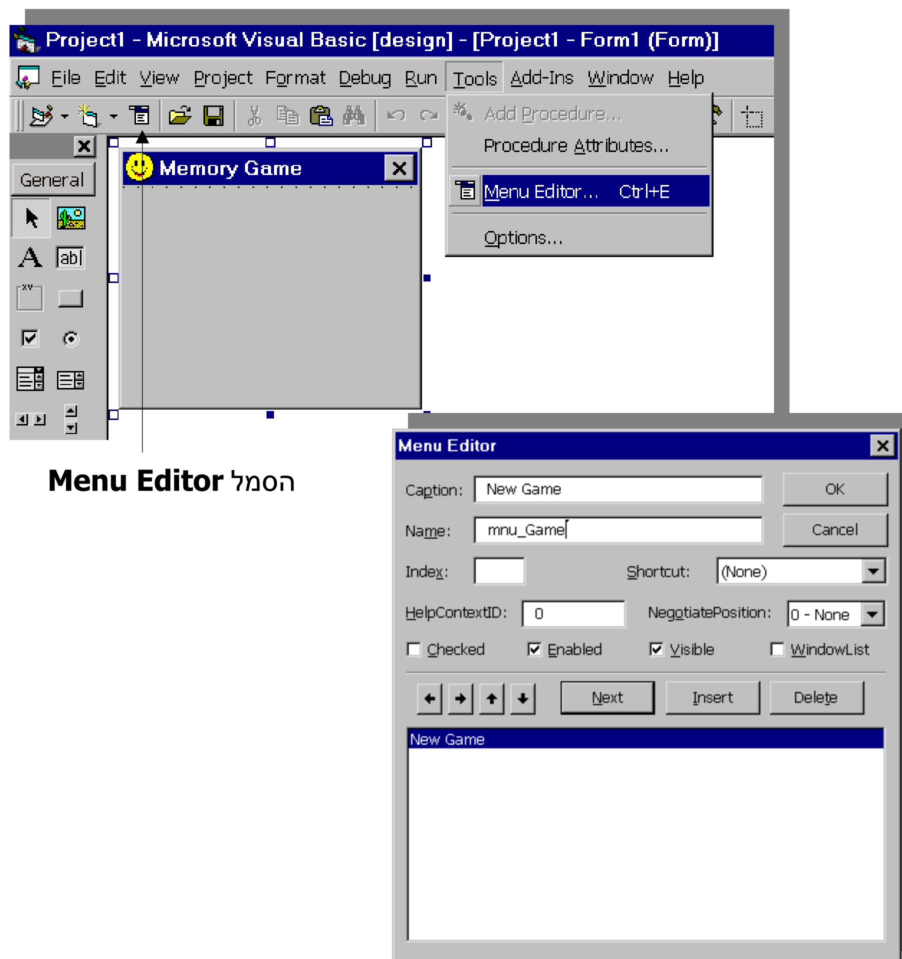
```
Private Sub Timer1_Timer()
    Form1.Caption = "Your time is " & i
      If i = 15 Then
            Frame1.Visible = True
            Label1.Caption = "Time is over"
            Timer1.Enabled = False
      End If
      If cmd_Face1.Enabled = False And _
            cmd_Face2.Enabled = False And _
            cmd_Face3.Enabled = False And _
            cmd_Face4.Enabled = False And _
            cmd_Face5.Enabled = False And _
            cmd_Face6.Enabled = False And _
            cmd_Face7.Enabled = False And _
            cmd_Face8.Enabled = False Then
```

```
                Frame1.Visible = True
                Label1.Caption = "Your time is " & i
                Timer1.Enabled = False
        End If
    i = i + 1
End Sub
```

יוצרים את התפריט

ניתן ליצור למשחק שלנו מספר רמות קושי, יחד עם האפשרות להתחיל את המשחק מחדש. כל זאת ניתן לביצוע **בעורך התפריט** (**Menu Editor**).

כדי לפתוח את עורך התפריט, יש להיכנס לחלון עיצוב התוכנית, לסמן את הטופס בעזרת העכבר, לפתוח את תפריט **Tools** ולבחור את הפריט **Menu Editor**, או לחילופין - ללחוץ על סמלון, אשר בשורה העליונה.

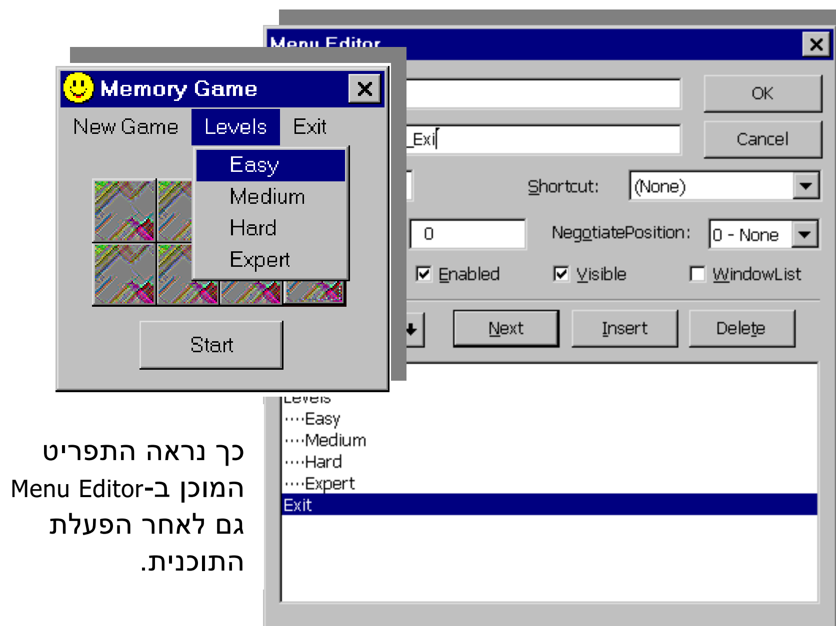


הסמל **Menu Editor**

1. בתיבת הטקסט **Caption** נקליד את כותרת התפריט - **New Game**, ובתיבת הטקסט **Name** - את שם התפריט - **mnu_Game** (**mnu** מקובל ב- Visual Basic כקיצור לשמות תפריט).
2. נלחץ על הלחצן **Next** ונוסיף את התפריט **Levels** עם השם **mnu_Levels**.
3. שוב נלחץ על **Next** ולאחר מכן על הלחצן עם החץ ימינה. נוסיף את הרמה הראשונה - **Easy** עם השם **itm_Easy** (**itm** קיצור המילה **Items**, המקובל ב-Visual Basic להצגת פריטי תפריט).
4. בדרך זו נוסיף את הפריטים הבאים:

Caption - כותרת	Name - שם
Easy	**itm_Easy**
Medium	**Itm_Medium**
Hard	**Itm_Hard**
Expert	**Itm_Expert**

5. נלחץ על הלחצן **Next** ולאחר מכן על הלחצן עם החץ שמאלה. נוסיף את התפריט האחרון: **Exit - mnu_Exit**.
6. כאשר נסיים נלחץ על **OK**.



כך נראה התפריט המוכן ב-Menu Editor גם לאחר הפעלת התוכנית.

קביעת רמות קושי שונות במשחק

את משך המשחק ניתן לנהל בעזרת המאפיין **Interval**. משום שקבענו שמספר "התקתוקים" של הטיימר שווה ל-15, המרווח בין "התקתוקים" מגדיר את משך המשחק הכללי (נזכיר שוב, שהמרווח נמדד באלפיות השנייה, ושנייה אחת שווה ל-1000 אלפיות השנייה).

1. בחלון עיצוב הטופס נפתח את התפריט ונלחץ על הרמה הראשונה - **Easy**, כדי להיכנס לחלון הקוד, נקליד:

```
Private Sub mnu_Easy_Click()
        Timer1.Interval = 1500    ' זמן המשחק 22.5 שניות
End Sub
```

2. כעת נכתוב את השגרות לכל יתר הרמות:

```
Private Sub mnu_Medium_Click()
       Timer1.Interval = 1000     ' זמן המשחק 15 שניות
End Sub

Private Sub mnu_Hard_Click()
       Timer1.Interval = 750      ' זמן המשחק 11.25 שניות
End Sub

Private Sub mnu_Expert_Click()
       Timer1.Interval = 500      ' זמן המשחק 7.5 שניות
End Sub
```

עכשיו ננסה לשחק במשחק שלנו, על ידי בחירת רמות הקושי השונות ונראה שברמות **Hard** או **Expert** לא כל כך קל לנצח במשחק בפעם הראשונה!

תוכנית ל- New Game

1. כדי להתחיל משחק חדש, יש להחזיר למרכיבים את התכונות שהיו להם בהתחלה. למטרה זו ניכנס לחלון הקוד לשיגרה **mnu_Game**.

2. את הלחצנים **cmd_Face1 ...cmd_Face8** נהפוך לבלתי גלויים.

```
       cmd_Face1.Visible = False
       cmd_Face2.Visible = False
       cmd_Face3.Visible = False
       cmd_Face4.Visible = False
       cmd_Face5.Visible = False
       cmd_Face6.Visible = False
       cmd_Face7.Visible = False
       cmd_Face8.Visible = False
```

3. למרות שהם בלתי גלויים, הם צריכים להיות "חיים" בתחילת המשחק, עד שנצליח להוציא אותם מהמשחק. לכן לכל הלחצנים נכתוב:

```
        cmd_Face1.Enabled = True
        cmd_Face2.Enabled = True
        cmd_Face3.Enabled = True
        cmd_Face4.Enabled = True
        cmd_Face5.Enabled = True
        cmd_Face6.Enabled = True
        cmd_Face7.Enabled = True
        cmd_Face8.Enabled = True
```

4. גם את הלחצנים-"מכסים" צריך להחזיר ל"חיים".

```
        cmd_Card1.Enabled = True
        cmd_Card2.Enabled = True
        cmd_Card3.Enabled = True
        cmd_Card4.Enabled = True
        cmd_Card5.Enabled = True
        cmd_Card6.Enabled = True
        cmd_Card7.Enabled = True
        cmd_Card8.Enabled = True
```

5. ולהפוך אותם לגלויים:

```
        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
```

6. נסיר את המסגרת (**Frame**), אחרת הלחצנים לא ייראו:

```
Frame1.Visible = False
```

7. נאפס את מונה הטיימר:

```
i = 0
```

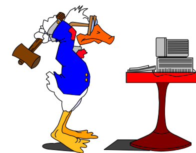
8. ונחזיר את הכיתוב הקודם לשורת הכותרת:

```
Form1.Caption = "Memory game"
```

השלמת הפריטים

כל תוכנית דורשת את ההשלמה. בחברות המפתחות תוכניות מחשב יש עובדים מיוחדים שמבצעים בדיקות של התוכניות המוכנות ומגלים את "הבאגים" הנסתרים. משום שאין לנו עובדים כאלה, נצטרך לעשות את בדיקת התוכנית בעצמנו.

"באג" מס' 1



נתחיל מזה, שאפשר לשחק גם בלי הלחצן **Start**, כלומר לא להדליק את הטיימר בזמן המשחק. נעשה כך, שהמסגרת (**Frame1**) תסתיר את הלחצנים עד שאנו נבחר רמת משחק כלשהי.

1. נסמן בעכבר את המסגרת (**Frame1**) ונשנה את המידות שלה כך שהלחצן **Start** יהיה גלוי.
2. נהפוך את לחצן **Start** לנסתר (מאפיין **Visible** - **False**). מעכשיו לא ניתן ללחוץ עליו עד שנבחר את רמת המשחק.
3. כדי שהשחקן יידע שיש לבחור קודם רמה, נקליד **Caption** לתווית **Label1** - **Select your level**.
4. המסגרת (**Frame1**) עם התווית (**Label1**) שעליה, צריכות להיות גלויות בתחילת המשחק. לכן נגדיר ל-**Frame1** את המאפיין **Visible** - **True**.
5. לאחר בחירת הרמה, צריך להופיע הלחצן **Start**, והכיתוב על התווית צריך להיות מוסתר. באמצעות העכבר נפתח את חלון הקוד לפריט התפריט **itm_Easy**, ונוסיף לשיגרה את שתי הפקודות:

```
cmd_Start.Visible = True
Label1.Caption = " "       ' מוחקים את הכיתוב מהתווית
```

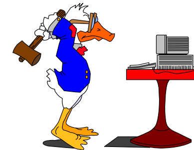
6. נקליד את הפקודות לשאר פריטי התפריט (**itm_Medium**, **itm_Hard** ו-**itm_Expert**).
7. הלחצן **Start** צריך להסתיר את המסגרת (**Frame**), ולפתוח את הלחצנים-"מכסים". בשיגרה ל-**cmd_Start** נכתוב:

```
Frame1.Visible = False
```

8. ושוב נסתיר את הלחצן הזה, כדי שלא ניתן יהיה ללחוץ עליו עד שתיבחר רמת הקושי במשחק:

```
cmd_Start.Visible = False
```

"באג" מס' 2



מתברר שבאמצע המשחק השחקן יכול ללחוץ בעכבר על התפריט **Levels**. דבר זה יגרום להופעתו הבלתי רצויה של הלחצן **cmd_Start**. לכן עדיף לכבות תפריט זה ולעשות את זה ברגע שמפעילים את לחצן **Start**.

1. לשגרת **cmd_Start** נוסיף את הפקודה הבאה:

```
mnu_Levels.Enabled = False
```

2. נחזיר את התפריט ל"חיים" רק כאשר מפעילים את המשחק החדש. לכן בשגרת **mnu_Game** צריך להוסיף:

```
mnu_Levels.Enabled = True
```

"באג" מס' 3



בזמן המשחק אסור ללחוץ בעכבר על התפריט **New Game**, הפעולה תפעיל משחק חדש אך עם שארית חישוב הזמן של המשחק הקודם.

1. נסמן בעכבר את הטופס **Form1**. נפתח את חלון **Menu Editor**, ונסיר את סימן הבדיקה (**CheckBox**) - "✓" מהאפשרות **Enabled** של התפריט **New Game**. עכשיו אי-אפשר להיכנס לתפריט זה בתחילת המשחק.

2. נגרום לכך שהתפריט יהיה נגיש רק לאחר סיום המשחק. לשם כך נפתח את שגרת הטיימר ונקליד שם את הפקודה **mnu_Game.Enabled = True** פעמיים: פעם ראשונה כאשר הזמן נגמר:

```
If i = 15 Then
        Frame1.Visible = True
        Label1.Caption = "Time is over"
        Timer1.Enabled = False
        mnu_Game.Enabled = True
End If
```

3. ופעם שנייה אם הצלחנו לגלות את כל התמונות:

```
If cmd_Face1.Enabled = False And _
        cmd_Face2.Enabled = False And _
        cmd_Face3.Enabled = False And _
        cmd_Face4.Enabled = False And _
        cmd_Face5.Enabled = False And _
        cmd_Face6.Enabled = False And _
        cmd_Face7.Enabled = False And _
```

```
        cmd_Face8.Enabled = False Then
        Frame1.Visible = True
        Label1.Caption = "Your time is " & i
        Timer1.Enabled = False
        mnu_Game.Enabled = True
End If
```

4. לאחר שבחרנו את התפריט **New Game**, נהפוך אותו מייד לבלתי נגיש, כלומר לשגרת **mnu_Game** נוסיף את הפקודה הבאה:

```
mnu_Game.Enabled = False
```

בחירת הרמות - השלב האחרון

כדי לסמן את הרמה הנבחרת אפשר להשתמש בתכונה **Checked**.

1. נכניס תוספת לשגרת הפריט **itm_Easy**:

```
itm_Easy.Checked = True
```

2. נפעיל את התוכנית ונבדוק כיצד התכונה הזאת עובדת ברמה **Easy**: הסימון "✓" - **Cheked** - צריך להופיע על הרמה הנבחרת.
3. נכתוב פקודות **itm_Medium.Checked = True**, **itm_Hard.Checked = True** ו- **itm_Expert.Checked = True** עבור שאר הרמות.
4. מעכשיו, אם ניכנס לתפריט **Levels** ונתחיל לבחור רמה אחרי רמה, כולן תהיינה מסומנות (**Checked**) על ידי הסימון "✓".
5. כדי להסיר את הסימונים משאר הרמות, יש להוסיף לשגרות את הפקודות:

```
Private Sub itm_Easy_Click()
        Timer1.Interval = 1500
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = True           ' מסומן
        itm_Medium.Checked = False        ' לא מסומן
        itm_Hard.Checked = False          ' לא מסומן
        itm_Expert.Checked = False        ' לא מסומן
End Sub

Private Sub itm_Medium_Click()
        Timer1.Interval = 1000
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False           ' לא מסומן
```

```
        itm_Medium.Checked = True          ' מסומן
        itm_Hard.Checked = False           ' לא מסומן
        itm_Expert.Checked = False         ' לא מסומן
End Sub

Private Sub itm_Hard_Click()
        Timer1.Interval = 750
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False           ' לא מסומן
        itm_Medium.Checked = False         ' לא מסומן
        itm_Hard.Checked = True            ' מסומן
        itm_Expert.Checked = False         ' לא מסומן
End Sub

Private Sub itm_Expert_Click()
        Timer1.Interval = 500
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False           ' לא מסומן
        itm_Medium.Checked = False         ' לא מסומן
        itm_Hard.Checked = False           ' לא מסומן
        itm_Expert.Checked = True          ' מסומן
End Sub
```

אם כל השגרות נרשמו נכון, הסימון **Checked** צריך להישאר רק על פריט התפריט הנבחר.

המשחק נגמר

נשאר לנו רק לכתוב את שגרת היציאה מהמשחק - **Exit**:

```
Private Sub mnu_Exit_Click()
        End
End Sub
```

הגענו לסוף המשחק שלנו. אם כל השגרות נרשמו נכון - המשחק צריך לעבוד! אם משהו אינו מסתדר, אפשר להשוות את מה שכתבנו עם טקסט התוכנית המלא שמופיע בדפים הבאים.

הטקסט המלא של התוכנית "משחק הזיכרון"

```
Dim i As Integer

Private Sub cmd_Card1_Click()
        cmd_Focus.SetFocus
        cmd_Card1.Visible = False
        cmd_Face1.Visible = True
        If cmd_Face1.Visible = True And _
        cmd_Face2.Visible = True Then
          cmd_Face1.Enabled = False
          cmd_Face2.Enabled = False
          cmd_Card1.Enabled = False
          cmd_Card2.Enabled = False
        End If
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub

Private Sub cmd_Card2_Click()
        cmd_Focus.SetFocus
        cmd_Card2.Visible = False
        cmd_Face2.Visible = True
        If cmd_Face1.Visible = True And _
          cmd_Face2.Visible = True Then
                cmd_Face1.Enabled = False
                cmd_Face2.Enabled = False
                cmd_Card1.Enabled = False
                cmd_Card2.Enabled = False
        End If
```

```
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub

Private Sub cmd_Card3_Click()
        cmd_Focus.SetFocus
        cmd_Card3.Visible = False
        cmd_Face3.Visible = True
        If cmd_Face3.Visible = True And _
          cmd_Face4.Visible = True Then
                cmd_Face3.Enabled = False
                cmd_Face4.Enabled = False
                cmd_Card3.Enabled = False
                cmd_Card4.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card4_Click()
        cmd_Focus.SetFocus
        cmd_Card4.Visible = False
        cmd_Face4.Visible = True
        If cmd_Face3.Visible = True And _
          cmd_Face4.Visible = True Then
                    cmd_Face3.Enabled = False
                    cmd_Face4.Enabled = False
                    cmd_Card3.Enabled = False
                    cmd_Card4.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card5_Click()
        cmd_Focus.SetFocus
        cmd_Card5.Visible = False
        cmd_Face5.Visible = True
        If cmd_Face5.Visible = True And _
          cmd_Face6.Visible = True Then
                    cmd_Face5.Enabled = False
                    cmd_Face6.Enabled = False
                    cmd_Card5.Enabled = False
                    cmd_Card6.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False
```

```
        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card7.Visible = True
    cmd_Card8.Visible = True
End Sub

Private Sub cmd_Card6_Click()
        cmd_Focus.SetFocus
        cmd_Card6.Visible = False
        cmd_Face6.Visible = True
        If cmd_Face5.Visible = True And _
          cmd_Face6.Visible = True Then
                    cmd_Face5.Enabled = False
                    cmd_Face6.Enabled = False
                    cmd_Card5.Enabled = False
                    cmd_Card6.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
End Sub
```

```
Private Sub cmd_Card7_Click()
        cmd_Focus.SetFocus
        cmd_Card7.Visible = False
        cmd_Face7.Visible = True
        If cmd_Face7.Visible = True And _
          cmd_Face8.Visible = True Then
                    cmd_Face7.Enabled = False
                    cmd_Face8.Enabled = False
                    cmd_Card7.Enabled = False
                    cmd_Card8.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
End Sub
```

```
Private Sub cmd_Card8_Click()
        cmd_Focus.SetFocus
        cmd_Card8.Visible = False
        cmd_Face8.Visible = True
        If cmd_Face7.Visible = True And _
          cmd_Face8.Visible = True Then
                    cmd_Face7.Enabled = False
                    cmd_Face8.Enabled = False
                    cmd_Card7.Enabled = False
                    cmd_Card8.Enabled = False
        End If
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
```

```
        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
End Sub
Private Sub cmd_Start_Click()
        Timer1.Enabled = True
        Frame1.Visible = False
        cmd_Start.Visible = False
        mnu_Levels.Enabled = False
End Sub

Private Sub itm_Easy_Click()
        Timer1.Interval = 1500
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = True
        itm_Medium.Checked = False
        itm_Hard.Checked = False
        itm_Expert.Checked = False
End Sub
Private Sub itm_Expert_Click()
        Timer1.Interval = 500
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False
        itm_Medium.Checked = False
        itm_Hard.Checked = False
        itm_Expert.Checked = True
End Sub
Private Sub itm_Hard_Click()
        Timer1.Interval = 750
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False
        itm_Medium.Checked = False
        itm_Hard.Checked = True
        itm_Expert.Checked = False
End Sub
```

```
Private Sub itm_Medium_Click()
        Timer1.Interval = 1000
        cmd_Start.Visible = True
        Label1.Caption = ""
        itm_Easy.Checked = False
        itm_Medium.Checked = True
        itm_Hard.Checked = False
        itm_Expert.Checked = False
End Sub
```

```
Private Sub mnu_Exit_Click()
        End
End Sub
```

```
Private Sub mnu_Game_Click()
        cmd_Face1.Visible = False
        cmd_Face2.Visible = False
        cmd_Face3.Visible = False
        cmd_Face4.Visible = False
        cmd_Face5.Visible = False
        cmd_Face6.Visible = False
        cmd_Face7.Visible = False
        cmd_Face8.Visible = False

        cmd_Face1.Enabled = True
        cmd_Face2.Enabled = True
        cmd_Face3.Enabled = True
        cmd_Face4.Enabled = True
        cmd_Face5.Enabled = True
        cmd_Face6.Enabled = True
        cmd_Face7.Enabled = True
        cmd_Face8.Enabled = True

        cmd_Card1.Enabled = True
        cmd_Card2.Enabled = True
        cmd_Card3.Enabled = True
        cmd_Card4.Enabled = True
        cmd_Card5.Enabled = True
        cmd_Card6.Enabled = True
        cmd_Card7.Enabled = True
        cmd_Card8.Enabled = True

        cmd_Card1.Visible = True
        cmd_Card2.Visible = True
```

```
        cmd_Card3.Visible = True
        cmd_Card4.Visible = True
        cmd_Card5.Visible = True
        cmd_Card6.Visible = True
        cmd_Card7.Visible = True
        cmd_Card8.Visible = True
        i = 0
        Frame1.Visible = False
        Form1.Caption = "Memory game"
        mnu_Levels.Enabled = True
        mnu_Game.Enabled = False
End Sub
```

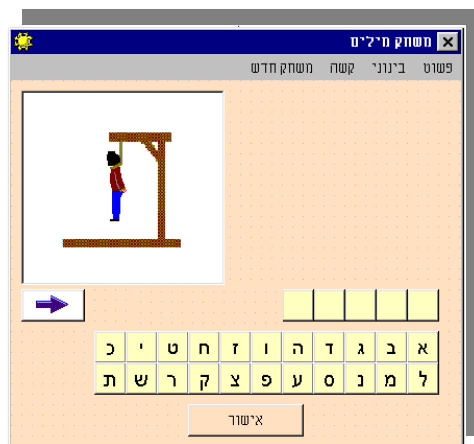
```
Private Sub Timer1_Timer()
        Form1.Caption = "Your time is " & i
        If i = 15 Then
          Frame1.Visible = True
          Label1.Caption = "Time is over"
          Timer1.Enabled = False
          mnu_Game.Enabled = True
        End If
        If cmd_Face1.Enabled = False And _
          cmd_Face2.Enabled = False And _
                cmd_Face3.Enabled = False And _
                cmd_Face4.Enabled = False And _
                cmd_Face5.Enabled = False And _
                cmd_Face6.Enabled = False And _
                cmd_Face7.Enabled = False And _
                cmd_Face8.Enabled = False Then
                Frame1.Visible = True
                Label1.Caption = "Your time is " & i
                Timer1.Enabled = False
                mnu_Game.Enabled = True
        End If
        i = i + 1
End Sub
```

3 משחק מילים "התליין"

תוכן הפרויקט

יש משחק ישן ומשעשע, שנקרא "התליין": מנחה המשחק בוחר מילה ורושם אותה על דף. השחקנים צריכים לנחשה, כל אחד בתורו. כשאחד מהשחקנים מנחש אות שגויה, המנחה מצייר חלק מעמוד התלייה על דף השחקן. בשגיאה הבאה הוא משלים את העמוד. בשגיאות הבאות הוא מצייר את ראשו, ידיו ורגליו של המסכן.

כאשר השחקן מנחש אות נכונה, מנהל המשחק מציין את מיקומה במילה (לא משנה היכן היא נמצאת).



המשחק מסתיים כאשר מישהו מנחש את כל המילה, או כאשר כל השחקנים עולים על בימת הגרדום.

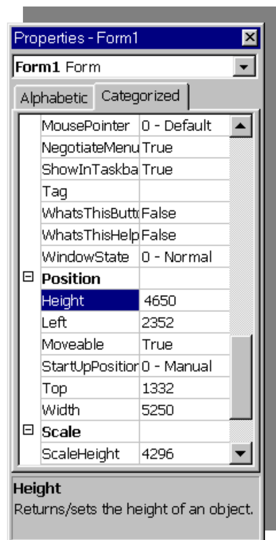
כללי המשחק

המשחק מורכב משדה המשחק, שעליו נמצאים לחצנים להקלדת המילים. מספר האותיות במילה תלוי ברמת המשחק (קל - 3 אותיות, בינוני - 4 וקשה - 5).

השחקן הראשון לוחץ בעזרת העכבר בלחצן-האות הרצוי לו, ובצורה כזו הוא מקליד מילה על "הלוח". אם האות מוקלדת אינה הרצויה, הוא לוחץ על לחצן **BackSpace**. כאשר המילה הוקלדה, הוא לוחץ על לחצן **אישור**, "הלוח" עם המילה נעלם, במקומו מופיע אותו "לוח", אבל עם המשבצות ריקות בצבע אחר.

השחקן השני מתייחס מול המחשב ומנסה לנחש את המילה באמצעות לחיצת עכבר על מקשי "המדלקלת".

השלב הראשון בביצוע הפרויקט שלנו הוא יצירת הגירסה השלמה של התוכנית באנגלית. הסיבה לכך היא, שהקפיצה המתמדת מעברית לאנגלית (כשאנגלית היא שפת התכנות) גורמת לקשיים נוספים בעבודה על הפרויקט. לאחר סיום כתיבת התוכנית, נוכל לתרגם בקלות את שפת המשחק מאנגלית לעברית.



1. נקבע מידות לחלון העבודה (**Form1**) : רוחב (**Width**) 5250 וגובה (**Height**) 4650. עדיף לקבוע את מידות **Width** ו-**Height** בחלון המאפיינים (**Properties**) ולא לגרור את מסגרת החלון באמצעות העכבר.

2. נשנה את המאפיין **Caption** של הטופס (**Form**) ל- **Word Game** ונמצא לתוכנית את הסמל המתאים (מאפיין **Icon**).

כדי ששתי המידות **Width** ו-**Height** תופענה בחלון **Properties** קרובות אחת לשנייה, עדיף להשתמש במיון מאפיינים לפי סוג (**Categorized**) ולא לפי סדר אלפאביתי (**Alphabetic**).

בניית חלון עבודה של התוכנית

1. בארגז הכלים נבחר את המרכיב (**CommandButton**) ונצייר לחצן במידות **Width =372**, **Height = 372**.

2. נקרא ללחצן בשם cmd_A (cmd - קיצור שם לחצן פקודה המקובל ב- Visual Basic).

3. בשורת מאפיינים (**Caption**) נקליד את האות A.

4. נשנה את המאפיין **Style** ל-**Graphical** ונצבע את הלחצן בצבע ירוק בהיר (מאפיין **BackColor**, אפשרות **Palette**).

5. נסמן את לחצן cmd_A באמצעות העכבר ונבחר להעתיקו (**Copy**).

6. נבחר את הפקודה **Paste**. על השאלה : "**You already have a control named 'cmd_A'. Do you want to create a control array?**" נענה - **No**.

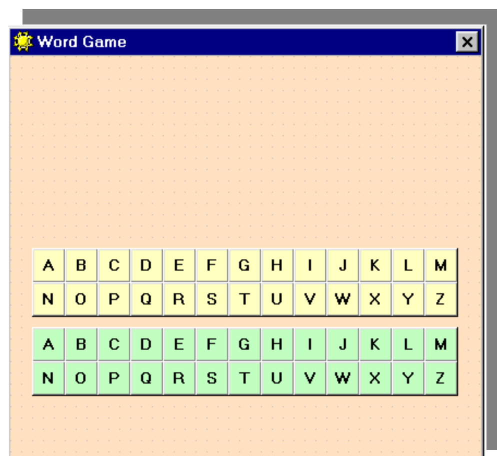
7. נשנה את שם הלחצן החדש ל-cmd_B ואת מאפיין **Caption** ל-**B**.

8. נחזור על הפעולה לכל 26 אותיות האלפאבית האנגלי.

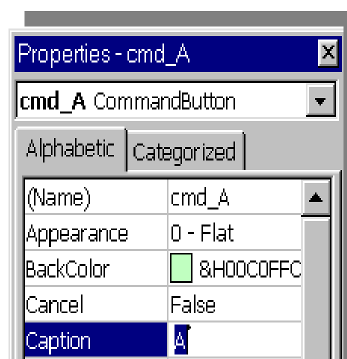
9. ניצור לחצן כזה נוסף ונקרא לו **c_A**.

10. נקבע ללחצן זה את מאפיין **Style - Graphical** ונצבע אותו בצבע צהוב בהיר.

11. בדרך זו ניצור 26 לחצנים-עותקים בשמות **c_A ... c_Z**.



12. באמצעות העכבר נסמן את כל הלחצנים הצהובים ונקבע את מאפיין Visible - False. בתחילת המשחק לחצנים אלה צריכים להיות מוסתרים.



ובכן, יצרנו שתי "מקלדות" - ירוקה וצהובה. "המקלדת" הירוקה משמשת להקלדת המילה על ידי השחקן הראשון ו"המקלדת" הצהובה משמשת לניחוש המילה על ידי השחקן השני. הפרוצדורות המנהלות את פעולת הלחצנים בכל המקלדת תהיינה שונות לחלוטין, לכן נאלצנו לצבע את כל ההכנות.

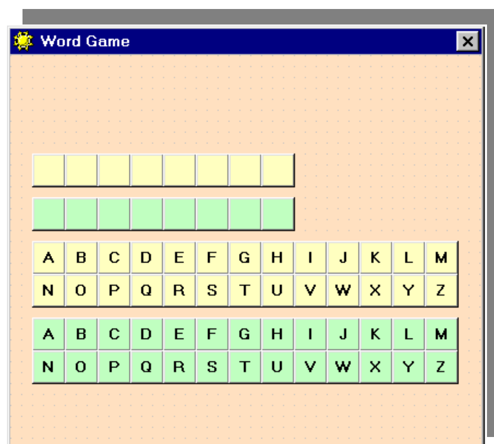
לאחר יצירת "המקלדות" יש לבדוק האם "פספסנו" שם לחצן כלשהו. למען מטרה זו, יש ללחוץ על כל לחצן זה אחר זה: שם הלחצן יופיע במסגרת העליונה של חלון המאפיינים.

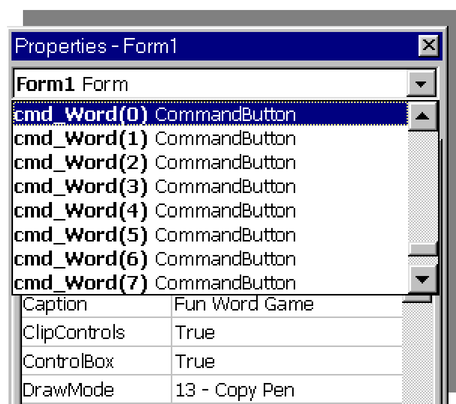
בניית לחצני "לוח"

1. ניצור לחצן נוסף בצבע ירוק בהיר. את מאפיין **Caption** נשאיר ריק, במילים אחרות - על הלחצן לא כתוב שום דבר. נקרא ללחצן זה בשם **cmd_Word**.

2. נעתיק אותו (**Copy**), אחר נבחר את הפקודה **Paste**, ועל השאלה: "**You already have a control named 'cmd_A'. Do you want to create a control array?**" נענה **Yes**.

3. עכשיו צריך ליצור שמונה לחצנים בצבע ירוק **cmd_Word**. לחצנים אלה משמשים כ"לוח" להקלדת המילה. חשוב שהלחצנים לא יהיו לחצנים בודדים, אלא מערך (**Array**). לכולם יש שם זהה - **cmd_Word**, אך הם נבדלים באינדקס (**Index**) שנמצא בין הסוגריים לאחר שם כל לחצן, למשל **cmd_Word(1)**, או **cmd_Word(5)**.





ב-Visual Basic מספור מרכיבי המערך מתחיל מאפס באופן אוטומטי, לכן במערך שלנו הלחצן הראשון נקרא **cmd_Word(0)** והלחצן האחרון נקרא **cmd_Word(7)** ולא **cmd_Word(8)**!



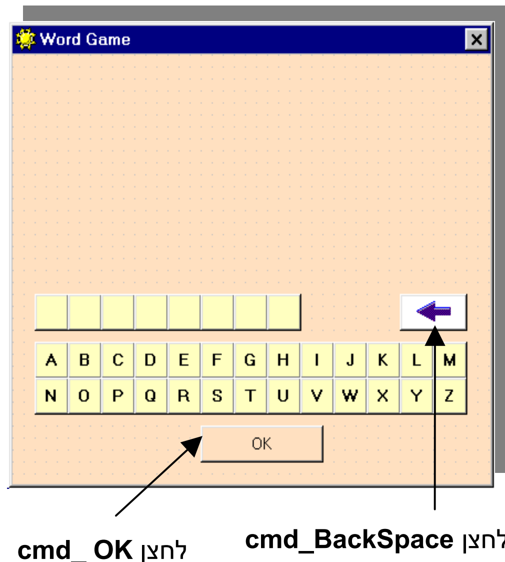
4. כעת, עלינו לבנות מערך הזה אשר מורכב מ-8 לחצנים צהובים ולקרוא להם **cmd_True**. לחצנים אלה משמשים לניחוש המילה.

5. נסמן בעכבר את שני המערכים: **cmd_Word** ו-**cmd_True** ונקבע את מאפיין **Visible - False**.

לחצנים נוספים

1. עכשיו עלינו לאסוף בעכבר את כל לחצני "המדלקת" הצהובה ולשים אותם על פני הלחצנים הירוקים. נעשה את אותה הפעולה עם לחצני החולה.

2. נצטרך שני לחצנים נוספים - **cmd_BackSpace** ו-**cmd_OK**.
ל-**cmd_BackSpace** אפשר לבחור את תמונת של חץ כמאפיין **Picture**. לחצן **BackSpace** משמש למחיקת אותיות שגויות ולחצן **OK** - לסיום הקלדת המילה. לאחר לחיצה עליו "המדלקת" הירוקה ו"החולה" הירוק ייעלמו, ובמקומם יופיעו "המדלקת" הצהובה ו"החולה" הצהוב.



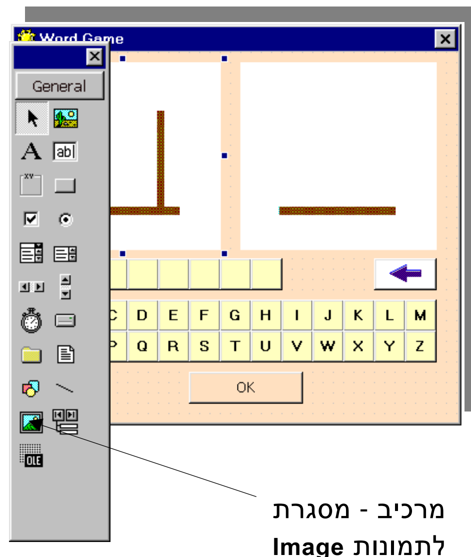
לחצן **cmd_BackSpace**   לחצן **cmd_ OK**

התקנת תמונות

1. נוסיף לטופס את מרכיב **Image**. נשתמש בתמונה הראשונה - **#1.bmp** (בתיקיה Images) כמאפיין **Picture**.

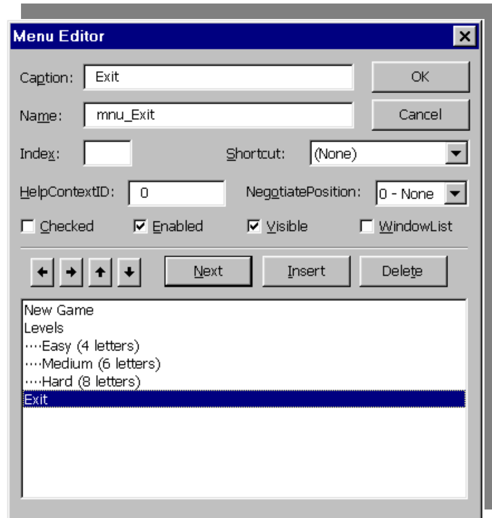
2. בעזרת הפקודות **Copy** ו-**Paste** ניצור מערך של 6 מסגרות: **Image1(0)...Image1(5)**. בדיוק באותה דרך שיצרנו מערכי לחצנים **cmd_Word** ו-**cmd_True**. בכל אחת מהמסגרות הנוצרות נוסיף קבצי תמונות **#2.bmp...#6.bmp**.

3. נסמן בעכבר את כל מרכיבי **Image1** ונקבע בשבילם **Left = 2600** ו-**Top =120**.



מרכיב - מסגרת לתמונות **Image**

יציאת התפריט

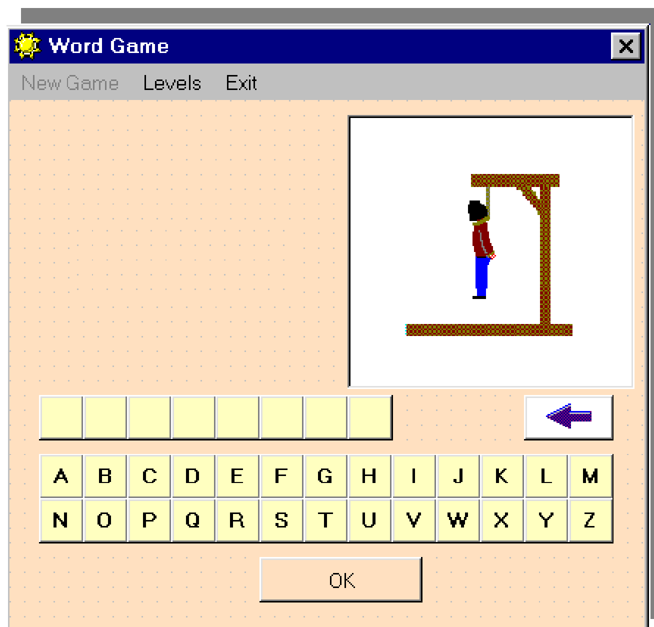


1. נפתח את עורך התפריטים (**Menu Editor**) וניצור תפריטים, כפי שנראה בתמונה.

2. לתפריטים נקרא בשמות הבאים:

Easy (4 letters) - itm_Easy
Medium (6 letters) - itm_Medium
Hard (8 letters) - itm_Hard
Exit - mnu_Exit

3. לתפריט **New Game** נסיר את התכונה **Enabled**, זאת אומרת נעביר אותו למצב **Enabled - False**.



הטופס שלנו מוכן לחלונית לתכנות המשחק.

הגדרת משתנים

קודם כל אנו צריכים לבחור את רמת המשחק - מספר האותיות במילה שלנו. למטרה זו נגדיר את המשתנה הראשון שלנו - מונה האותיות **Let_Counter**.

```
Dim Let_Counter as Integer
```

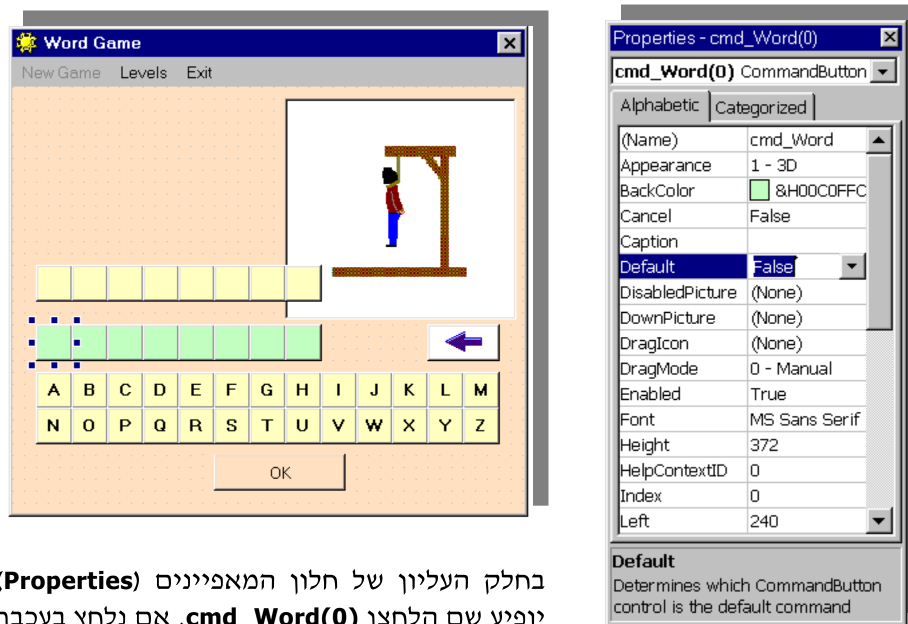
חוץ מזה נצטרך מונה נוסף למרכיבי המערך. בתכנות, נהוג לתת למונים אלה שמות-אותיות: **i**, **j**, **k** וכולי.

```
Dim i as Integer
```

פרוצדורת בחירת רמת המשחק

בזמן יצירת התפריט כבר הגדרנו שברמת "**Easy**" המילה כוללת 4 אותיות, ברמת "**Medium**" - 6 אותיות וברמת "**Hard**" - 8 אותיות. זאת אומרת, לאחר לחיצה על כל אחד מפריטים אלה צריכים להופיע על המסך 4, 6 או 8 מלחצני "החולה" **cmd_Word**.

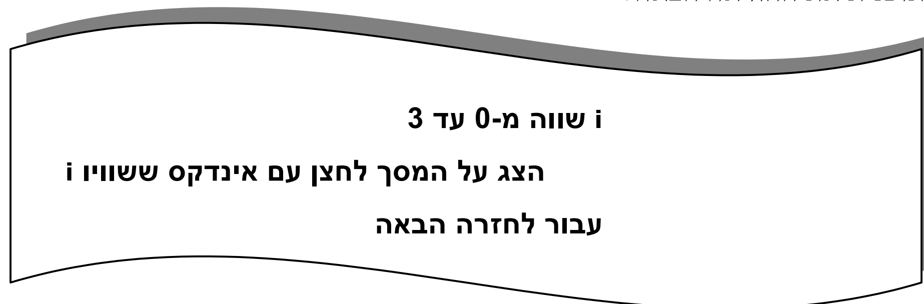
נזיז לרגע את לחצני **cmd_True** ונפשר גישה ללחצני **cmd_Word** שנמצאים מתחתם. לאחר מכן, נסמן את הלחצן הראשון באמצעות העכבר.



בחלק העליון של חלון המאפיינים (**Properties**) יופיע שם הלחצן **cmd_Word(0)**. אם נלחץ בעכבר על שאר הלחצנים, נוכל לקרוא את שמותיהם: כולם נקראים **cmd_Word**, אך נבדלים במספר שבסוגריים - אינדקס (**Index**). לחצנים אלה יוצרים מערך (**Array**) ומקבלים מספרים לפי הסדר, החל מאפס.

כדי להציג אחד מלחצני המערך, למשל, לחצן שלו אינדקס **3**, נכתוב את הפקודה **cmd_Word(3).Visible = True**, והלחצן הרביעי יופיע. שוב, יש לזכור שמספור מרכיבי המערך מתחיל מאפס, לכן אינדקס **3** מתייחס ללחצן הרביעי ולא לשלישי.

תכנות מרבית הפעולות על מערכים (**Arrays**) מתבצע בעזרת לולאות (**Loops**). כיצד עובדת לולאה? נגיד, שאנו צריכים להציג על המסך 4 לחצני "לוח" (רמת **Easy**). נתאים למשתנה **Let_Count** ערך **3** (זכרו, שמספור מתחיל **מאפס** ולא **מ-1**!) וניתן לתוכנית את ההוראה הבאה:



עכשיו נראה כיצד אפשר לרשום את הלולאה הזו בשפת Visual Basic:

```
For i = 0 To 3
   cmd_Word(i).Visible = True
Next
```

בתחילת הלולאה משתנה i שווה לאפס. נציין שמשתנה זה מהווה כעת את מונה-האינדקס שבשם של לחצני **cmd_Word(i)**. כלומר, החזרה (איטרציה) הראשונה של הלולאה תתרחש כאשר **i = 0** . לכן, הלולאה מתחילה במילים **For i = 0**.

אם כן, מה יקרה כתוצאה מביצוע הפקודות בחזרה הראשונה? אם נשנה את **i** לאפס בפקודה המתבצעת, נקבל

```
cmd_Word(0).Visible = True
```

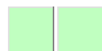
ועל המסך יופיע לחצן "הלוח" הראשון.

בחזרה הבאה, כאשר **i = 1**, יופיע הלחצן השני ואחריו השלישי והרביעי עד שהלולאה תיעצר, הרי התנינו את תנאי הביצוע שלה: רק כשערך **i** הוא מאפס עד **Let_Count**. (**For** i = 0 **To** Let_Count) ו-**Let_Count** ברמת **Easy** שווה **3**.

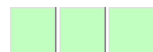
החזרה הראשונה, i = 0 cmd_Word(0).Visible = True



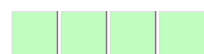
החזרה השנייה, i = 1 cmd_Word(1).Visible = True



החזרה השלישית, i = 2 cmd_Word(2).Visible = True



החזרה הרביעית, i = 3 cmd_Word(3).Visible = True



הפרוצדורה שנכתוב לרמת **Easy** תיראה כך:

```
Private Sub itm_Easy_Click()
     Let_Count = 3
        For i = 0 To Let_Count
          cmd_Word(i).Visible = True
        Next
   End Sub
```

כעת נבדוק כיצד עובדת התוכנית שלנו. נחזיר את לחצני **cmd_True** למקומם ונפעיל את התוכנית בעזרת לחצן **Start**. נפתח את תפריט **Levels** ונבחר את רמת **Easy**. על המסך יופיעו ארבעה לחצני "הלוח". נבחר את הרמה הבאה, ויופיעו שישה לחצנים. ברמת **Hard** התוכנית "מציירת" לנו שמונה לחצנים. אך מה קורה? אנו רוצים לחזור לרמת **Easy**, אך מספר הלחצנים אינו יורד! איננו יכולים להכריח את התוכנית לפעול בכיוון ההפוך, זאת אומרת, היא יכולה להוסיף לחצני "לוח", אך אינה יכולה להורידם, בינתיים!

כדי שהתוכנית תעבוד גם בסדר הפוך, אפשר להשתמש במין "טריק". לפני שהתוכנית מתחילה לצייר את לחצני ה"לוח", כל הלחצנים שקיימים כעת, ללא חשיבות למספר, נמחקים. אפילו אם אנו מפעילים את התוכנית בפעם הראשונה - לא נורא: ניקוי המסך לא יימשך זמן רב. גם לא חשוב כמה לחצנים מוצגים על המסך - נמחק את כל השמונה (עד **i = 7**)!

לצורך ניקוי זה נשתמש בלולאה דומה ללולאה הקודמת, כשההבדל היחיד הוא שמאפיין **Visible** הוא **False** ולא **True**:

```
For i = 0 To 7
      cmd_Word(i).Visible = False
Next
```

נוסיף קטע זה לתחילת כל אחת משלושת הפרוצדורות של **Easy**, **Medium** ו-**Hard**, ויחד עם זאת, בסוף כל פרוצדורה נחזיר את **i** להתחלה: **i = 0** (לאחר מכן נסביר מדוע אנו עושים זאת).

```
Private Sub itm_Easy_Click()
        For i = 0 To 7
            cmd_Word(i).Visible = False
        Next
        Let_Count = 3
        For i = 0 To Let_Count
          cmd_Word(i).Visible = True
        Next
        i = 0
End Sub
```

```
Private Sub itm_Medium_Click()
	For i = 0 To 7
	  cmd_Word(i).Visible = False
	Next
	Let_Count = 5
	For i = 0 To Let_Count
	  cmd_Word(i).Visible = True
	Next
	i = 0
End Sub
```

```
Private Sub itm_Hard_Click()
	For i = 0 To 7
	  cmd_Word(i).Visible = False
	Next
	Let_Count = 7
	For i = 0 To Let_Count
	  cmd_Word(i).Visible = True
	Next
	i = 0
End Sub
```

הקלדת המילה

עכשיו, כשיש לנו "לוח" ו"מקלדת", אפשר לכתוב את הפרוצדורה להקלדת המילה. זו לא תהיה פרוצדורה אחת, אלא 26 פרוצדורות בודדות (**Private Sub**) המיועדות לכל לחצן של "המקלדת". כמובן, כל פרוצדורות הלחצנים תהיינה זהות בכל, פרט לכך שכל לחצן יקליד לתוך "הלוח" את האות שלו. לכן מספיק להקליד את הפרוצדורה הבאה ללחצן הראשון **cmd_A**, ולאחר מכן להעתיק אותה לשאר הלחצנים.

כך נראה ההליך (פרוצדורה):

```
Private Sub cmd_A_Click()
	cmd_Word(i).Caption = "A"
	i = i + 1
	cmd_BackSpace.Visible = True
End Sub
```

נגיד שלחצנו בעכבר על הלחצן **A** (**cmd_A**) של "המקלדת". מה יקרה?

הפקודה **cmd_Word(i).Caption = "A"** שולחת את האות **A** לתוך "הלוח", ללחצן שהאינדקס שלו שווה i. ניזכר, שבסוף כל פרוצדורה של בחירת הרמה כתבנו את הפקודה **i=0**. בדרך זו ציינו את תחילת המילה, הלחצן הראשון של "הלוח" -

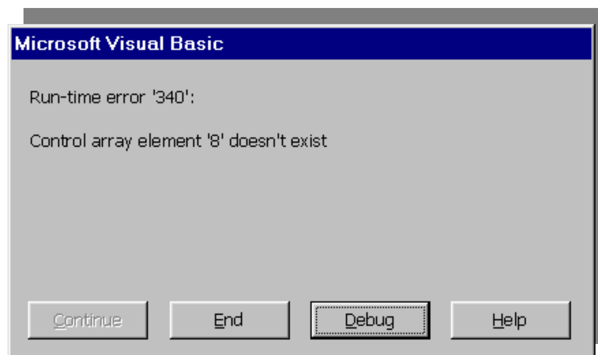
cmd_Word(0), ובלחצן זה תוקלד האות של הלחצן הראשון (במקרה זה **cmd_A**).
כיצד נתקדם ללחצן "הלוח" הבא? נוסיף פקודה להגדלת האינדקס ב-1 (**i = i + 1**), ונוכל להקליד את האות השנייה.

אם טעינו והקלדנו אות שגויה, נזדקק ללחצן **BackSpace** כדי למחוק אותה. את פרוצדורת המחיקה נכתוב מאוחר יותר, בינתיים יש להסיר לחצן זה מהמסך, כי כל לחיצה עליו לפני שהאות הראשונה הוקלדה, תגרום להודעת שגיאה בתפקוד התוכנית (כמובן, לאחר שפרוצדורת **cmd_BackSpace** תיכתב). לכן, יש לקבוע בחלון המאפיינים של לחצן זה את ערך המאפיין **Visible - False**.

כעת משמעות הפקודה הבאה תהיה מובנת: **cmd_BackSpace.Visible = True**.
כשנקליד את האות הראשונה על "הלוח", לחצן **BackSpace** יהפוך מ-**Visible - False** ל-**Visible - True**.

נראה כיצד עובדת פרוצדורה זו. למשל, נבחר את רמת **Hard** ונקליד ל"לוח" את האות **A** שמונה פעמים. הצלחתם?

כעת נלחץ על **cmd_A** פעם נוספת. מייד תופיע על המסך:

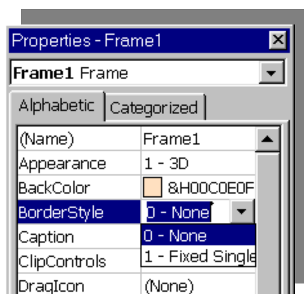


אין צורך לפחוד מהודעות אלו וללחוץ מייד על אחד הלחצנים. הודאה זו עוזרת לנו לאתר שגיאה בתוכנית שלנו. נראה מה משמעות המשפט הבא:

"Control array element '8' doesn't exist"

יצרנו מערך (**Array**) מ-8 מרכיבים-לחצנים **cmd_Word**. מספור המרכיבים מתחיל מאפס ומסתיים באינדקס שערכו 7. כלומר, המרכיב השמיני אינו קיים במערך שלנו, ולכן קיבלנו הודעת שגיאה. לחיצה נוספת בעכבר תפעיל את הפקודה **i = i + 1**, כשערך **i** כבר שווה 7, ותנסה לרשום את האות **A** ללחצן **cmd_Word (8)**, שאינו קיים בטופס שלנו.

איך נמנע את הניסיון הזה?



נצייר את המסגרת (**Frame**) על פני לחצני "המקלדת". בחלון המאפיינים נבחר את המאפיין **BorderStyle - 0 - None** ולמאפיין **Visible** נקבע את הערך **False**.

כעת המסגרת (**Frame**) פשוט תסגור (תמנע) את הגישה ללחצני "המקלדת" לאחר שתוקלדנה אותיות לתוך כל לחצני "הלוח":

```
If i > Let_Count Then
          Frame1.Visible = True
End If
```

להלן הטקסט המלא להליך הלחצן **cmd_A**:

```
Private Sub cmd_A_Click()
        cmd_Word(i).Caption = "A"
        i = i + 1
        cmd_BackSpace.Visible = True
        If i > Let_Count Then
                    Frame1.Visible = True
        End If
End Sub
```

כעת נותר רק להעתיק את ההליך ולהוסיפו לכל לחצנים: **cmd_B**, **cmd_C** וכולי.

חשוב לא לשכוח לשנות את האותיות בשורה: **Caption = "A"** בהתאמה לשם הלחצנים ולבדוק את פעולת התוכנה. אם כל הפקודות כתובות נכון, התוכנה שלנו תרשה להקליד לתוך לחצני "הלוח" את כל 26 האותיות של האלפאבית האנגלי.

מחיקת אותיות

הלחצן **BackSpace**, שהסרנו מהמסך (למטרה זו הגדרנו את המאפיין שלו **Visible** כ-**False**), נחוץ לנו כעת כדי למחוק את האותיות השגויות. לחצן זה יופיע על המסך כאשר האות הראשונה הוקלדה (אולי, נרצה למחוק אותה!). כל אחת מ-26 האותיות יכולה להיות הראשונה, לכן, הפקודה שמציגה את לחצן **BackSpace** על המסך, כלולה בהליך כל לחצן מ-**cmd_A** עד **cmd_Z**.

תפקוד הלחצן עצמו מתואר בהליך הבא:

```
Private Sub cmd_BackSpace_Click()
        cmd_Word(i - 1).Caption = ""
        i = i - 1
        If i = 0 Then
           cmd_BackSpace.Visible = False
        End If
        Frame1.Visible = False
End Sub
```

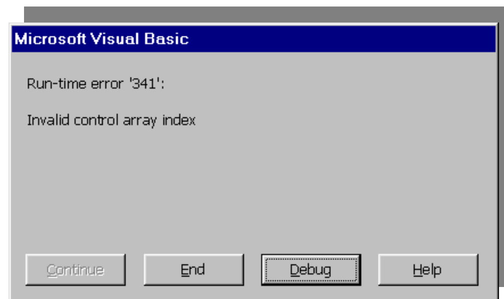
כדי להבין את משמעות ההליך, נראה מה קרה למשתנה **i** לאחר הקלדת האות הבאה. נזדקק רק לשתי פקודות המסומנות במסגרת:

```
Private Sub cmd_A_Click()
        cmd_Word(i).Caption = "A"
        i = i + 1
        cmd_BackSpace.Visible = True
        If i > Let_Count Then
                Frame1.Visible = True
        End If
End Sub
```

דיברנו כבר על כך שבכל פעם שאנו מקלידים אות, המונה **i** גדל ב-1 ומאפשר להתקדם ללחצן הבא של "הלוח". למשל, אם A תהיה האות שהוקלדה ראשונה, אז **i** היה שווה לאפס לפני הלחיצה על הלחצן **cmd_A**, והופך להיות שווה ל-1 לאחר הלחיצה. כלומר, כדי למחוק את האות, צריך ***לחזור אליה*** (**i - 1**) ולכתוב פקודת מחיקה. מחיקה מתבצעת באמצעות מתן ערך ריק למאפיין **Caption**, כך: **Caption = " "**.

בדרך זו אפשר למחוק כל אות ב"לוח" ולא רק את האות הראשונה, אך כדי לחזור לאות הקודמת, יש להקטין את **i** ב-1 (**i = i - 1**).

כשהאות הראשונה של "הלוח" תימחק ו-**i** יהיה שווה לאפס, לא תישארנה אותיות למחיקה, וניסיון ללחוץ על הלחצן **BackSpace** יוביל להודעה הבאה.



כדי להימנע מהודעה זו, נשתמש בדרך המוכרת לנו כבר - כאשר אין מה למחוק יותר, נסיר את הלחצן מהמסך:

```
If i = 0 Then
        cmd_BackSpace.Visible = False
End If
```

ייתכן גם מצב שנזדקק ללחצן **BackSpace** למחיקת האות האחרונה. במקרה זה תיווצר בעיה, משום שכתבנו פרוצדורה ש"סוגרת" את "המקלדת" לאחר הקלדת האות האחרונה, ולכן נוכל למחוק אותיות, אך נוכל להקליד אותיות חדשות במקומן?

לא קשה לפתור בעיה זו. נקבע שכל לחיצה על **cmd_BackSpace** תסיר מהפרוצדורה את המסגרת המסתירה את "המקלדת".

```
Frame1.Visible = False
```

השחקנים מתחלפים ביניהם

ובכן, השחקן הראשון סיים להקליד את המילה שלו. עכשיו הוא צריך להסתיר את "הלוח" הירוק שבו מופיעה המילה שלו, ולהציג במקומו את ה"לוח" הצהוב שבו מספר לחצנים זהה, כדי שהשחקן השני יתחיל לנחש את המילה. פרוצדורה זו מתבצעת בעזרת לחצן **cmd_OK**, שמופיע ברגע שהאות האחרונה ב"לוח" הוקלדה, זאת אומרת, התנאי **i > Let_Count** התבצע. נוסיף את הפרוצדורות הבאות ללחצני "המקלדת" מ- **cmd_A** עד **cmd_Z**:

```
Private Sub cmd_A_Click()
        cmd_Word(i).Caption = "A"
        i = i + 1
        cmd_BackSpace.Visible = True
        If i > Let_Count Then
          Frame1.Visible = True
          cmd_OK.Visible = True
        End If
End Sub
```

שימו לב למיקום הפקודה החדשה: **cmd_OK.Visible = True**.

לחצן **OK** ייבצע את הפקודות הבאות:

1. בלולאה מ-**0** עד ל-**Let_Count** צריכים להופיע הלחצנים הצהובים **cmd_True** במקום הלחצנים הירוקים **cmd_Word**:

```
For i = 0 To Let_Count
   cmd_Word(i).Visible = False
   cmd_True(i).Visible = True
Next
```

2. הלחצן **BackSpace** אינו נחוץ לנו כבר - נסיר אותו:

```
cmd_BackSpace.Visible = False
```

3. נסיר את המסגרת (**Frame**) שמסתירה לנו את "המקלדת":

```
Frame1.Visible = False
```

4. התפריט **Levels** יפריע לנו כעת מכיון שמישהו יכול ללחוץ עליו במקרה, ולקלקל את כל המשחק:

```
mnu_Levels.Enabled = False
```

5. לא נזדקק גם ללחצן **OK**:

```
cmd_OK.Visible = False
```

6. את המונה **i** יש להחזיר לערך ההתחלתי:

```
i = 0
```

7. כל אותיות "המקלדת" הירוקה מ-**cmd_A** עד **cmd_2**, צריכות להיעלם:

```
cmd_A.Visible = False
....................................................
cmd_Z.Visible = False
```

8. במקומם יופיעו לחצני "המקלדת" הצהובה מ-**c_A** עד ל-**c_Z**:

```
c_A.Visible = True
....................................................
c_Z.Visible = True
```

להלן הטקסט המלא של הפרוצדורה:

```
Private Sub cmd_OK_Click()
        For i = 0 To Let_Count
          cmd_Word(i).Visible = False
          cmd_True(i).Visible = True
        Next
        cmd_BackSpace.Visible = False
        Frame1.Visible = False
        mnu_Levels.Enabled = False
        cmd_OK.Visible = False
        i = 0
        cmd_A.Visible = False
        cmd_B.Visible = False
        cmd_C.Visible = False
        cmd_D.Visible = False
        cmd_E.Visible = False
        cmd_F.Visible = False
        cmd_G.Visible = False
        cmd_H.Visible = False
        cmd_I.Visible = False
        cmd_J.Visible = False
        cmd_K.Visible = False
        cmd_L.Visible = False
        cmd_M.Visible = False
        cmd_N.Visible = False
        cmd_O.Visible = False
        cmd_P.Visible = False
        cmd_Q.Visible = False
        cmd_R.Visible = False
        cmd_S.Visible = False
        cmd_T.Visible = False
        cmd_U.Visible = False
        cmd_V.Visible = False
        cmd_W.Visible = False
        cmd_X.Visible = False
        cmd_Y.Visible = False
        cmd_Z.Visible = False

        c_A.Visible = True
        c_B.Visible = True
        c_C.Visible = True
        c_D.Visible = True
        c_E.Visible = True
        c_F.Visible = True
        c_G.Visible = True
        c_H.Visible = True
```

```
        c_I.Visible = True
        c_J.Visible = True
        c_K.Visible = True
        c_L.Visible = True
        c_M.Visible = True
        c_N.Visible = True
        c_O.Visible = True
        c_P.Visible = True
        c_Q.Visible = True
        c_R.Visible = True
        c_S.Visible = True
        c_T.Visible = True
        c_U.Visible = True
        c_V.Visible = True
        c_W.Visible = True
        c_X.Visible = True
        c_Y.Visible = True
        c_Z.Visible = True
End Sub
```

נבדוק את התוצאה של עבודת הלחצן **OK**: על המסך יופיע טופס עם "המקלדת" הצהובה ו"הלוח" הצהוב. לחצני **BackSpace** ו-**OK** ייעלמו, ולחלק התפריט **Levels** אין גישה (**Enabled = False**).

עכשיו, ננסה לנחש את המילה שהוקלדה.

ניחוש מילה

כדי לנחש את המילה, השחקן לוחץ על לחצני "המקלדת" הצהובה: מ-**c_A** עד ל-**c_Z**. כשנבין את עיקרון עבודת התוכנית ונכתוב את הפרוצדורה ללחצן **c_A**, נעתיק את הפרוצדורה לכל שאר הלחצנים ונשנה את האותיות בהתאם.

ובכן, מה יקרה כאשר השחקן השני ילחץ על הלחצן **c_A**? יש כאן שתי אפשרויות:

- **אפשרות 1** (נקרא לה מצב **True**): האות **A** נמצאת במילה שלנו.
- **אפשרות 2** (נקרא לה מצב **False**): האות **A** אינה נמצאת במילה שלנו.

נדבר על האפשרות כשהאות **A** נמצאת במילה שלנו. כעת מתעוררת השאלה, איפה היא נמצאת: בתחילת המילה, בסוף או באמצע. הרי לפי כללי המשחק, לא משנה, איפה האות נמצאת, אם שחקן ניחש אותה, מנהל המשחק מציב אותה במקומה. במשחק שלנו הוספנו שהיא גם צריכה להיעלם מ"המקלדת".

בואו נבדוק את כל לחצני **cmd_Word**, ונברר האם האות שהוקלדה מופיעה באחד מהם. לא מפריע לנו שהלחצנים נעלמו מהמסך (**cmd_Word (i).Visible = False**), התוכנה "זוכרת" את מה שהיה כתוב בהם.

```
For i = 0 To Let_Count
   If cmd_Word(i).Caption = "A" Then
     cmd_True(i).Caption = "A"
        End if
Next
```

נגיד שב"לוח" היו ארבע משבצות (**Let_Count = 3**) והאות **A** מוקמה במשבצת השנייה (**i = 1**). בלולאה **0** עד **3** נבדוק את התנאי **If cmd_Word(i).Caption = "A"**.

הלולאה מתחילה כאשר **i = 0**, בדרך זו אנו בודקים האם האות **A** קיימת במשבצת הראשונה. היא לא - התנאי If לא התבצע, והלולאה עוברת לחזרה השנייה (**i = 1**). כאן התוכנה מוצאת את האות **A**. אם התנאי **If** נכון, גם הפקודה המוגדרת לאחר המילה **Then** צריכה להתבצע: **cmd_True(i).Caption = "A"**. מובן, שהאות **A** נכנסת למשבצת השנייה של ה"לוח" הצהוב, כי בחזרה זו **i = 1**.

נוסיף ללולאה את תנאי היעלמות האות המנוחשת מ"המקלדת":

```
For i = 0 To Let_Count
        If cmd_Word(i).Caption = "A" Then
          cmd_True(i).Caption = "A"
          c_A.Visible = False
        End If
Next
```

עתה נזדקק למשתנה נוסף - מונה האותיות שנוחשו. נוסיף לחלק המשתנים של התוכנה שלנו (**General**) את המשתנה החדש:

```
Dim j As Integer
```

בכל פעם שהאות נוחשה נכון, נגדיל את המשתנה באחד.

```
For i = 0 To Let_Count
   If cmd_Word(i).Caption = "A" Then
     cmd_True(i).Caption = "A"
     c_A.Visible = False
        j = j + 1
   End If
 Next
```

נפתח את חלון העיצוב של הטופס שלנו ונוסיף תווית (**Label1**) למסגרת (**Frame1**). בחלון המאפיינים בשורת **Caption** נסיר את המילה **Label1**, נצבע את התווית באותו צבע שצבענו את הטופס והמסגרת (**Frame**) - במאפיין **BackColor**. במאפיין **Alignment** נבחר **Center**. אפשר לשנות גם את סוג הגופן, את גודלו (מאפיין **Font**) ואת הצבע שלו (מאפיין **ForeColor**). את המאפיין **Visible** נקבע כ-**False**.

כעת נוסיף לפרוצדורה שלנו את התנאי הבא: אם מספר האותיות המנוחשות (**j**) שווה למספר האותיות על "הלוח" (**Let_Count**) המקלדת תיעלם ועל המסגרת (**Frame**) תופיע תווית האומרת "ניצחת" ("You won!"). התנאי (**If**) השני צריך להיות בתוך התנאי (**If**) הראשון, **If** בתוך **If**. מבנים כאלה נקראים **IF מקונן** (**Nested If**), כאשר כל אחד מהם "נסגר" ב- **End If** משלו.

כדי למנוע מהשחקנים להחליף רמה באמצע המשחק, ננטרל את תפריט **Levels**. נוסיף את השורה: **Menu_levels.enabled = False** לאחר **End If** השני.

באותה הזדמנות, נוסיף גם את הפקודה המעירה את התפריט "**New Game**", כי לאחר ניצחון השחקן השני אפשר להתחיל את המשחק מחדש!

```
Private Sub c_A_Click()
For i = 0 To Let_Count
        If cmd_Word(i).Caption = "A" Then
          cmd_True(i).Caption = "A"
          c_A.Visible = False
           j = j + 1
                  If j > Let_Count Then
                          Frame1.Visible = True
                          Label1.Visible = True
                          Label1.Caption = "You won!"
                          mnu_New.Enabled = True
                  End If
        End If
Menu_levels.enabled = False
End Sub
```

כעת, אפשר להעתיק את הפרוצדורה הזו לכל השאר הלחצנים, להחליף את האות **A** לאותיות המתאימות ללחצנים, ולנסות לשחק במשחק. האמת, בינתיים המשחק עובד רק "לניחוש". אם לא ניחשת אות שום דבר לא קורה, ושום עמוד תלייה אינו נבנה. לעומת זאת, אם ניחשת נכון האות תופיע במקומה והמשחק נגמר בסופו בניצחון השחקן השני.

נגדיר בחלק **General** את המשתנה החדש **G** שיכול להיות רק **True** או **False**.
משתנים מסוג זה נקראים משתנים לוגיים ומוגדרים כך:

```
Dim G As Boolean
```

במצב הראשון, כאשר האות מנוחשת נכון, המשתנה **G** מקבל משמעות **G = True**.
נוסיף את השורה הזו לפרוצדורה המוכנה של כל הלחצנים הצהובים:

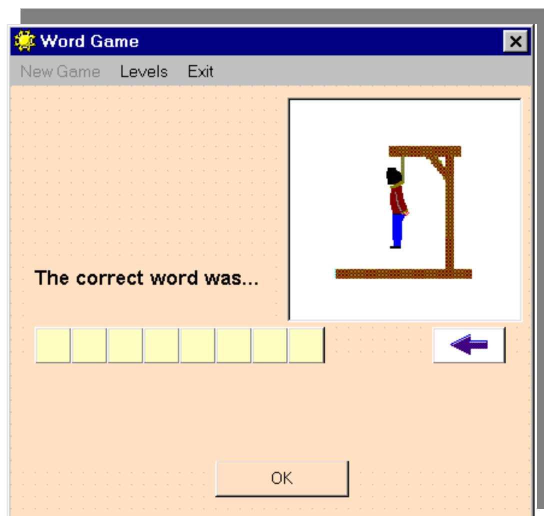
```
Private Sub c_A_Click()
For i = 0 To Let_Count
   If cmd_Word(i).Caption = "A" Then
     cmd_True(i).Caption = "A"
     c_A.Visible = False
     j = j + 1
        If j > Let_Count Then
           Frame1.Visible = True
           Label1.Visible = True
           Label1.Caption = "You won!"
           mnu_New.Enabled = True
        End If
      G = True
   End If
End Sub
```

כעת, נראה מה יקרה במצב השני כאשר **G = False**.

נגדיר בחלק **General** את המשתנה **k** - מונה ניסיונות הניחוש:

```
Dim k As Integer
```

נצייר תווית נוספת (**Label2**) על לחצני "הלוח". במאפיין **Caption** שלה נכתוב את המשפט "**The correct word was...**" ואת המאפיין **Visible** נגדיר כ-**False**.



נכתוב את החלק השני של הפרוצדורה:

```
If G = False Then
      Image1(k).Visible = True
      k = k + 1
        If k = 6 Then
          Frame1.Visible = True
          Label1.Visible = True
          Label1.Caption = "You dead!"
          Label2.Visible = True
                For i = 0 To Let_Count
                cmd_True(i).Visible = False
                cmd_Word(i).Visible = True
            Next
        End If
End If
G = False
```

ובכן כאשר האות מנוחשת לא נכון ו- **G = False** (אפשרות 2), מופיעה התמונה הראשונה - בסיס עמוד התלייה: **Image1(k).Visible = True**. בתחילת המשחק **k** שווה תמיד לאפס, לכן התמונה הראשונה מתאימה למרכיב המערך **Image1(0)**.

צריך להוסיף מייד למשתנה k אחד, כדי להכין את התמונה השנייה לקראת הכישלון הבא: **k = k + 1**.

כאשר נוצלו כל ששת ניסיונות הניחוש (**If k = 6 Then**), מופיעה מסגרת (**Frame1.Visible = True**) ועליה התווית (**Label1.Visible = True**) שעליה כתוב "את(ה) מת" (**Label1. Caption = "You dead!"**).

מעל משבצות "הלוח" תופיע הכתובת הרשומה בתווית **Label2**:

```
Label2.Visible = True
```

כל הלחצנים הצהובים של "הלוח" ייעלמו ובמקומם יופיע "הלוח" הירוק, שבו הקליד השחקן הראשון את המילה:

```
For i = 0 To Let_Count
    cmd_True(i).Visible = False
    cmd_Word(i).Visible = True
Next
```

בסוף הפרוצדורה נעניק למשתנה **G** את הערך **False**. המשתנה **G** יישאר עם ערך זה עד לניסיון הבא של השחקן השני לנחש את המילה. אם הניסיון עבר בצורה מוצלחת, פרוצדורת הלחצן הבא תעבוד לפי "**אפשרות 1 - True**". אם הניסיון נכשל, מייד יעבוד התנאי: **If G = False ...**

כעת נראה כיצד נראית הפרוצדורה המלאה של לחצני "המקלדת" הצהובה **c_A...c_Z**, ונעתיק את הטקסט לכל הלחצנים.

```
Private Sub c_A_Click()
For i = 0 To Let_Count
        If cmd_Word(i).Caption = "A" Then
        cmd_True(i).Caption = "A"
        c_A.Visible = False
        G = True
        j = j + 1
          If j > Let_Count Then
                    Frame1.Visible = True
                    Label1.Visible = True
                    Label1.Caption = "You won!"
                    mnu_New.Enabled = True
          End If
        End If
Next
        If G = False Then
          Image1(k).Visible = True
          k = k + 1
                    If k = 6 Then
                              Frame1.Visible = True
                              Label1.Visible = True
                              Label1.Caption = "You dead!"
                              Label2.Visible = True
                              mnu_New.Enabled = True
                                        For I = 0 To Let_Count
                                                  cmd_True(I).Visible = False
                                                  cmd_Word(I).Visible = True
                                                  mnu_New.Enabled = True
                                        Next
                    End If
        End If
G = False
End Sub
```

ננסה לשחק שוב

ראינו שגם במקרה ניצחון וגם לאחר "תלייה" אנו מעירים את חלק התפריט **New Game** (**mnu_New. Enabled = True**). זאת אומרת, מופיעה אפשרות להפעיל את המשחק מחדש. טקסט הפרוצדורה לחלק זה נראה כך:

```
Private Sub mnu_New_Click()
        mnu_New.Enabled = False
        mnu_Levels.Enabled = True

        Label2.Visible = False
        Frame1.Visible = True
        Label1.Visible = True
        Label1.Caption = "Select your level"

        c_A.Visible = False
        c_B.Visible = False
        c_C.Visible = False
        c_D.Visible = False
        c_E.Visible = False
        c_F.Visible = False
        c_G.Visible = False
        c_H.Visible = False
        c_I.Visible = False
        c_J.Visible = False
        c_K.Visible = False
        c_L.Visible = False
        c_M.Visible = False
        c_N.Visible = False
        c_O.Visible = False
        c_P.Visible = False
        c_Q.Visible = False
        c_R.Visible = False
        c_S.Visible = False
        c_T.Visible = False
        c_U.Visible = False
        c_V.Visible = False
        c_W.Visible = False
        c_X.Visible = False
        c_Y.Visible = False
        c_Z.Visible = False

        cmd_A.Visible = True
        cmd_B.Visible = True
        cmd_C.Visible = True
```

```
        cmd_D.Visible = True
        cmd_E.Visible = True
        cmd_F.Visible = True
        cmd_G.Visible = True
        cmd_H.Visible = True
        cmd_I.Visible = True
        cmd_J.Visible = True
        cmd_K.Visible = True
        cmd_L.Visible = True
        cmd_M.Visible = True
        cmd_N.Visible = True
        cmd_O.Visible = True
        cmd_P.Visible = True
        cmd_Q.Visible = True
        cmd_R.Visible = True
        cmd_S.Visible = True
        cmd_T.Visible = True
        cmd_U.Visible = True
        cmd_V.Visible = True
        cmd_W.Visible = True
        cmd_X.Visible = True
        cmd_Y.Visible = True
        cmd_Z.Visible = True
        For i = 0 To Let_Count
          cmd_Word(i).Visible = False
          cmd_True(i).Visible = False
          cmd_Word(i).Caption = ""
          cmd_True(i).Caption = ""
        Next
        For k = 0 To 5
          Image1(k).Visible = False
        Next
        i = 0
        k = 0
        j = 0
        G = False
End Sub
```

נעתיק את הטקסט לחלק **mnu_New** וננסה להבין כיצד עובדת הפרוצדורה. אין בה שום דבר חדש: כל הפקודות האלו כבר שימשו אותנו בפרויקט זה.

השלמת הפרטים

כפי שבפרויקט הקודם נשאר לנו להשלים מספר פרטים כדי שהתוכנית תעבוד ללא "באגים", כך גם בפרויקט זה.

1. נפעיל את המשחק. על המסך יופיע הטופס עם הלחצנים הירוקים **cmd_A ... cmd_Z**. כעת, נלחץ בעכבר על אחד מהם, וכתוצאה - כל האותיות תיעלמנה ויופיעו הלחצנים **cmd_BackSpace** ו-**cmd_OK**.
2. כדי למנוע לחיצה על לחצן עם אות לפני שמופיעים לחצני "הלוח" נהפוך את המסגרת (Frame1) לגלויה, כלומר, בחלון המאפיינים נקבע לה **True-Visible**.
3. לתווית Label1 נקליד כיתוב **Select your level** במאפיין **Caption**. הכיתוב הוא הנחיה לשחקן שבתחילת המשחק יש לבחור רמה.
4. אך הכיתוב אינו נעלם לאחר שבחרנו רמה, לכן עלינו להקליד את הפקודה הבאה לכל פרוצדורת בחירת רמות:

```
Frame1.Visible = False
```

במקרה זה המסגרת תיעלם יחד עם התווית שעליה, הלחצנים ייפתחו, ואפשר יהיה להתחיל במשחק.

5. כדי שהכיתוב לא יופיע יותר במהלך המשחק, נוסיף את הפקודה למחיקתו:

```
Label1.Caption = ""
```

המשחק נגמר

נכתוב את פרוצדורת היציאה מהמשחק - **Exit**:

```
Private Sub mnu_Exit_Click()
	End
End Sub
```

הקוד המלא של התוכנית "משחק מילים"

בקוד התוכנית הובאו פרוצדורות ללחצנים **cmd_A** ו-**c_A** בלבד. לשאר הלחצנים יש להחליף אות במקום המתאים, לכן אין צורך לחזור על הפרוצדורות בקוד התוכנית.

```
Dim Let_Count As Integer
Dim i,j,k As Integer
Dim G As Boolean


Private Sub cmd_OK_Click()
        For i = 0 To Let_Count
          cmd_Word(i).Visible = False
          cmd_True(i).Visible = True
        Next
        cmd_BackSpace.Visible = False
        cmd_OK.Visible = False
        Frame1.Visible = False
        mnu_Levels.Enabled = False
        i = 0

        cmd_A.Visible = False
        cmd_B.Visible = False
        cmd_C.Visible = False
        cmd_D.Visible = False
        cmd_E.Visible = False
        cmd_F.Visible = False
        cmd_G.Visible = False
        cmd_H.Visible = False
        cmd_I.Visible = False
        cmd_J.Visible = False
        cmd_K.Visible = False
        cmd_L.Visible = False
        cmd_M.Visible = False
        cmd_N.Visible = False
        cmd_O.Visible = False
        cmd_P.Visible = False
        cmd_Q.Visible = False
        cmd_R.Visible = False
        cmd_S.Visible = False
        cmd_T.Visible = False
        cmd_U.Visible = False
        cmd_V.Visible = False
        cmd_W.Visible = False
        cmd_X.Visible = False
        cmd_Y.Visible = False
```

```
        cmd_Z.Visible = False

        c_A.Visible = True
        c_B.Visible = True
        c_C.Visible = True
        c_D.Visible = True
        c_E.Visible = True
        c_F.Visible = True
        c_G.Visible = True
        c_H.Visible = True
        c_I.Visible = True
        c_J.Visible = True
        c_K.Visible = True
        c_L.Visible = True
        c_M.Visible = True
        c_N.Visible = True
        c_O.Visible = True
        c_P.Visible = True
        c_Q.Visible = True
        c_R.Visible = True
        c_S.Visible = True
        c_T.Visible = True
        c_U.Visible = True
        c_V.Visible = True
        c_W.Visible = True
        c_X.Visible = True
        c_Y.Visible = True
        c_Z.Visible = True
End Sub
```

```
Private Sub c_A_Click()
        For i = 0 To Let_Count
          If cmd_Word(i).Caption = "A" Then
                    cmd_True(i).Caption = "A"
                    c_A.Visible = False
                    G = True
                    J = J + 1
                              If J > Let_Count Then
                                        Frame1.Visible = True
                                        Label1.Visible = True
                                        Label1.Caption = "You won!"
                                        mnu_New.Enabled = True
                              End If
                    End If
        Next
          If G = False Then
                    Image1(k).Visible = True
```

```
                        k = k + 1
                                If k = 6 Then
                                        Frame1.Visible = True
                                        Label1.Visible = True
                                        Label1.Caption = "You dead!"
                                        Label2.Visible = True
                                                For i = 0 To Let_Count
                                                        cmd_True(i).Visible = False
                                                        cmd_Word(i).Visible = True
                                                        mnu_New.Enabled = True
                                                Next
                                End If
                End If
        G = False
End Sub
```

```
Private Sub cmd_A_Click()
        cmd_Word(i).Caption = "A"
        i = i + 1
        cmd_BackSpace.Visible = True
        If i > Let_Count Then
          Frame1.Visible = True
          cmd_OK.Visible = True
        End If
Menu_levels.enabled = False
End Sub
```

```
Private Sub cmd_BackSpace_Click()
        cmd_Word(i - 1).Caption = ""
        i = i - 1
        If i = 0 Then
          cmd_BackSpace.Visible = False
        End If
        Frame1.Visible = False
End Sub
```

```
Private Sub cmd_Confirm_Click()
        For i = 0 To Let_Count
          cmd_Word(i).Visible = False
          cmd_True(i).Visible = True
        Next
        cmd_BackSpace.Visible = False
        cmd_Confirm.Visible = False
        Frame1.Visible = False
        mnuEasy.Visible = False
        mnuMedium.Visible = False
        mnuHard.Visible = False
```

```
mnu_New.Visible = True
mnu_New.Enabled = False
i = 0
cmd_A.Visible = False
cmd_B.Visible = False
cmd_C.Visible = False
cmd_D.Visible = False
cmd_E.Visible = False
cmd_F.Visible = False
cmd_G.Visible = False
cmd_H.Visible = False
cmd_I.Visible = False
cmd_J.Visible = False
cmd_K.Visible = False
cmd_L.Visible = False
cmd_M.Visible = False
cmd_N.Visible = False
cmd_O.Visible = False
cmd_P.Visible = False
cmd_Q.Visible = False
cmd_R.Visible = False
cmd_S.Visible = False
cmd_T.Visible = False
cmd_U.Visible = False
cmd_V.Visible = False
cmd_W.Visible = False
cmd_X.Visible = False
cmd_Y.Visible = False
cmd_Z.Visible = False

c_A.Visible = True
c_B.Visible = True
c_C.Visible = True
c_D.Visible = True
c_E.Visible = True
c_F.Visible = True
c_G.Visible = True
c_H.Visible = True
c_I.Visible = True
c_J.Visible = True
c_K.Visible = True
c_L.Visible = True
c_M.Visible = True
c_N.Visible = True
c_O.Visible = True
c_P.Visible = True
c_Q.Visible = True
```

```
        c_R.Visible = True
        c_S.Visible = True
        c_T.Visible = True
        c_U.Visible = True
        c_V.Visible = True
        c_W.Visible = True
        c_X.Visible = True
        c_Y.Visible = True
        c_Z.Visible = True
End Sub

Private Sub mnu_Exit_Click()
        End
End Sub

Private Sub mnu_New_Click()
        mnu_New.Enabled = False
        mnu_Levels.Enabled = True
        Label2.Visible = False
        Frame1.Visible = True
        Label1.Visible = True
        Label1.Caption = "Select your level"

        c_A.Visible = False
        c_B.Visible = False
        c_C.Visible = False
        c_D.Visible = False
        c_E.Visible = False
        c_F.Visible = False
        c_G.Visible = False
        c_H.Visible = False
        c_I.Visible = False
        c_J.Visible = False
        c_K.Visible = False
        c_L.Visible = False
        c_M.Visible = False
        c_N.Visible = False
        c_O.Visible = False
        c_P.Visible = False
        c_Q.Visible = False
        c_R.Visible = False
        c_S.Visible = False
        c_T.Visible = False
        c_U.Visible = False
        c_V.Visible = False
        c_W.Visible = False
        c_X.Visible = False
```

```
        c_Y.Visible = False
        c_Z.Visible = False

        cmd_A.Visible = True
        cmd_B.Visible = True
        cmd_C.Visible = True
        cmd_D.Visible = True
        cmd_E.Visible = True
        cmd_F.Visible = True
        cmd_G.Visible = True
        cmd_H.Visible = True
        cmd_I.Visible = True
        cmd_J.Visible = True
        cmd_K.Visible = True
        cmd_L.Visible = True
        cmd_M.Visible = True
        cmd_N.Visible = True
        cmd_O.Visible = True
        cmd_P.Visible = True
        cmd_Q.Visible = True
        cmd_R.Visible = True
        cmd_S.Visible = True
        cmd_T.Visible = True
        cmd_U.Visible = True
        cmd_V.Visible = True
        cmd_W.Visible = True
        cmd_X.Visible = True
        cmd_Y.Visible = True
        cmd_Z.Visible = True

        For i = 0 To Let_Count
          cmd_Word(i).Visible = False
          cmd_True(i).Visible = False
          cmd_Word(i).Caption = ""
          cmd_True(i).Caption = ""
        Next
        For k = 0 To 5
          Image1(k).Visible = False
        Next
        i = 0
        k = 0
        J = 0
        G = False
End Sub

Private Sub itm_Easy_Click()
        For I = 0 To 7
```

```
        Cmd_word(I).visible
      Next
      Let_Count = 3
      For i = 0 To Let_Count
        cmd_Word(i).Visible = True
      Next
      i = 0
      Frame1.Visible = False
      Label1.Caption = ""
End Sub

Private Sub itm_Hard_Click()
      For I = 0 To 7
        Cmd_word(I).visible
      Next
      Let_Count = 7
      For i = 0 To Let_Count
        cmd_Word(i).Visible = True
      Next
      i = 0
      Frame1.Visible = False
      Label1.Caption = ""
End Sub

Private Sub itm_Medium_Click()
      For I = 0 To 7
        Cmd_word(I).visible
      Next
      Let_Count = 5
      For i = 0 To Let_Count
        cmd_Word(i).Visible = True
      Next
      i = 0
      Frame1.Visible = False
      Label1.Caption = ""
End Sub
```

 משחק "בינגו"

בפרויקט זה - משחק הילדים "בינגו", נכיר את פונקציית הרנדומיזציה (בחירה אקראית). פונקציות אלו מרשות לנו לכתוב תוכניות משחקים, שבהן המחשב הוא המשתתף השני במשחק.

יצירת ממשק המשחק

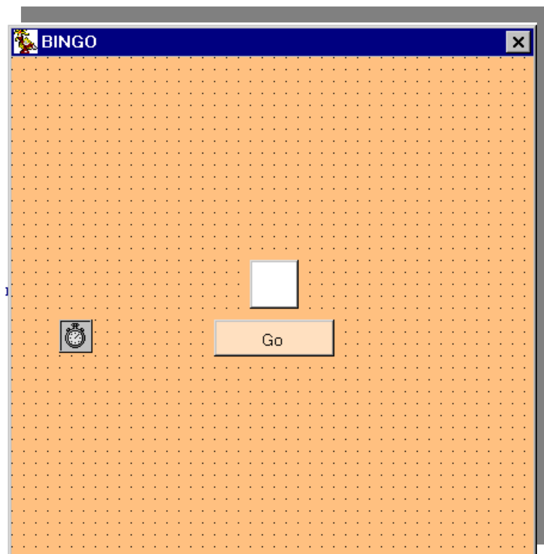
1. בסייר **Windows** ניצור תיקיה חדשה - **New Folder** ונקרא לה **Bingo**.
2. נעבור ל- **Visual Basic** ונפתח פרויקט חדש (**New Project**).
3. נקבע לטופס את המאפיינים הבאים:

- **Height - 5350**
- **Width - 5350**
- **BorderStyle - 1-Fixed Single**
- **Caption - Bingo**
- **Icon** - תמונה כלשהי מהתיקיה Icons שבתקליטור המצורף

אפשר לצבוע את הטופס בצבע נעים ושקט.

4. נוסיף לחצן (**CommandButton**) בגדלים ***492 X 492***, נקרא לו **cmd_ Show**, נגדיר את המאפיין **Style** כ-**Graphical** ונמחק את הכיתוב במאפיין **Caption**. נצבע את הלחצן בלבן (**BackColor**).
5. נוסיף לחצן נוסף, נקרא לו **cmd_Go** ונשנה את הכיתוב (**Caption**) - **Go**. אפשר לקבוע ללחצן אותו צבע (**BackColor**) שיש לטופס.

6. נוסיף לטופס טיימר - Timer1 ונגדיר את המאפיין של Enabled כ-False ואת המאפיין Interval - 1000.

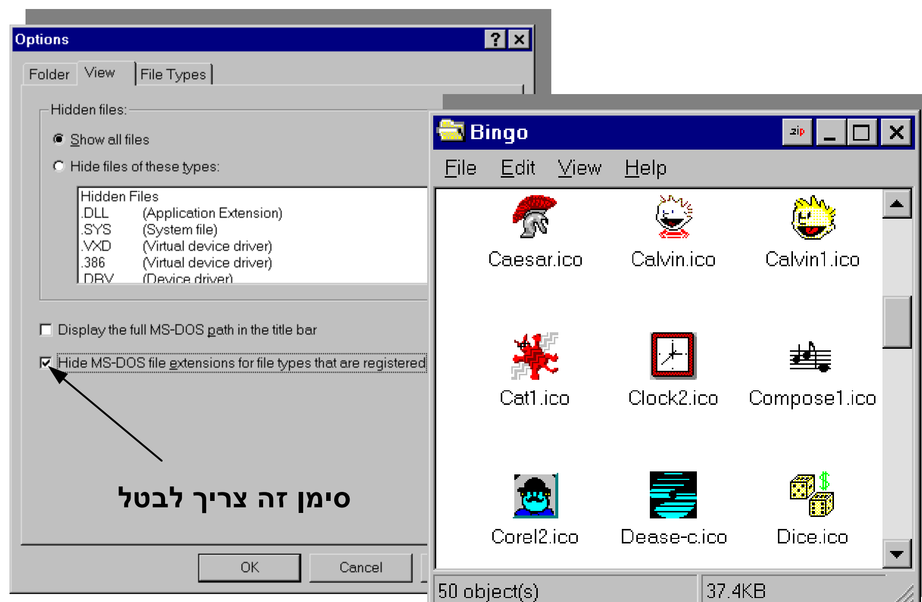


הכנת התמונות

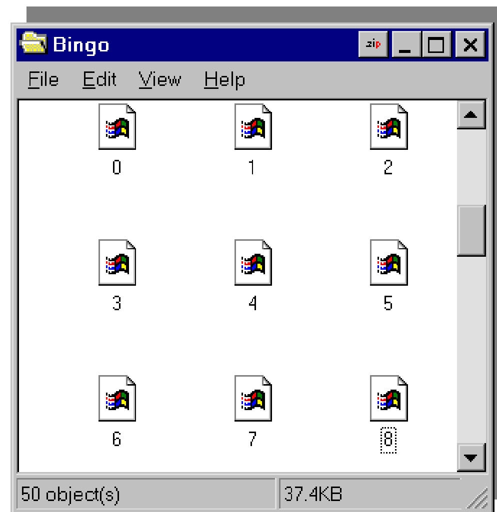
1. בסייר Windows נבחר 50 תמונות מהתיקייה Icons ונעתיק אותן לתיקיה Bingo.

2. בתפריט הראשי של התיקיה Icons נפתח את תפריט View (תצוגה) נבחר את הפריט Options (אפשרויות), בכרטיסייה View (תצוגה).

3. נסיר את הסימן ☑ שליד האפשרות **הסתר סיומות קבצים...** כדי שנוכל לראות את סיומות הקבצים, כפי שאפשר לראות בתמונה.



4. נחליף שמות לכל הקבצים, כדי שכולם יורכבו מהמספרים **0** עד **49** ללא סיומת **.ico** לצורך העניין נלחץ על שם הקובץ ואחר נלחץ עליו שוב (אך לא לחיצה כפולה - **Double Click**!) או, לחלופין, נלחץ על הלחצן הימני של העכבר בתפריט המקוצר שנפתח נבחר **Rename** (שנה שם). לאחר שינוי השמות, חלון התיקיה Bingo אמור להיראות כמו בתמונה זו.



דפדוף בין התמונות

במשחק Bingo יש "מסך" שעליו מופיעות תמונות בקצב קבוע. הלחצן **cmd_Show** ישמש בתוכניתנו מסך שכזה. אך התמונות אינן יכולות להופיע מעצמן. יש צורך במנגנון כלשהו כדי לפתוח תמונה, לסגור אותה, להוציא אותה מהמסך ולשים במקומה תמונה אחרת.

העיקרון הבסיסי של תפקוד התוכניות הכתובות בשפת Visual Basic הינו פעילות גומלין בין התוכנית למשתמש: התוכנית מצפה לפעולה של המשתמש. זה יכול להיות למשל, לחיצה על העכבר או הקשה על מקשי המקלדת, הזזת הסמן וכדומה. פעולות אלו נקראות ב- Visual Basic **אירועים** ומעובדות באמצעות תוכניות קטנות שנקראות **פרוצדורות** (או **הליך** או **שיגרה**). המרכיב היחיד שתפקודו אינו תלוי בפעולות המשתמש הוא הטיימר. הטיימר עובד כל זמן שמתבצע התנאי, הטיימר "דלוק" אם **Enabled - True**.

1. נפתח את פרוצדורת הלחצן **cmd_Go** ונכתוב את הפקודה שמפעילה את הטיימר:

```
Timer1. Enabled = True
```

2. בחלק העליון של חלון הקוד **(General)** נגדיר משתנה למספרי התמונות המתחלפים:

```
Dim Pic_Number As Integer
```

3. נכתוב את הפרוצדורה של הטיימר:

```
Private Sub Timer1_Timer()
        Pic_Number = Int(Rnd * 50)
        cmd_Show.Picture = LoadPicture(Pic_Number)
 End Sub
```

4. נשמור את הפרויקט ואת הטופס בתיקיה Bingo בשמות **bingo.vbp** ו-**bingo.frm** (חשוב שכל קבצי התמונות והפרויקט יימצאו באותה תיקיה!).

5. נסגור את חלון Visual Basic. נפתח את תיקיית Bingo, נאתר את קובץ **bingo.vbp** ונפתח אותו.

6. נפעיל את התוכנית. אם נלחץ כעת על לחצן **Go**, תתחלפנה התמונות שעל הלחצן-"מסך" (**cmd_Show**) באינטרוול של שנייה אחת.

כיצד התוכנית שלנו פועלת

בתיקיה שלנו נמצאים 50 קבצים-תמונות, שאת שמם וסוגם שינינו, מקבצים מסוג **ico.*** לקבצים ששמם הוא מספרים מ-0 עד 49. כעת אפשר להשתמש במספרי הקבצים כערכים שלמים (**Integer**) של המשתנה הרגיל **Pic_Counter**.

ב- Visual Basic המשמעות של **cmd_Show.Picture = LoadPicture (Pic_Counter)** היא:

התמונה שנקצה ללחצן cmd_Show בעזרת הפונקציה המיוחדת LoadPicture, נמצאת בקובץ ששמו שווה לערך המשתנה Pic_Counter.

כדי להבין כיצד הפקודה עובדת, נצייר זמנית על הטופס שלנו לחצן פקודה רגיל **Command1** ונקצה לו את הפקודה הבאה:

```
Private Sub Command1_Click()
   cmd_Show.Picture = LoadPicture(1)
End Sub
```

נלחץ על לחצן זה, ועל "המסך" תופיע תמונת הקובץ הנקרא 1. נשנה את שם (המספר) הקובץ, ונראה שעם כל לחיצה על לחצן **Command1**, תופיע על "המסך" תמונה חדשה. כעת אפשר למחוק את לחצן **Command1**.

כיצד אפשר לגרום לתמונות שעל "המסך" להתחלף באופן אוטומטי כל פרק זמן שווה, ומצד שני - שהופעת תמונה זו או אחרת תהיה מקרית? באמצעות ארגון התוכנית כך שהמשתנה **LoadPicture** יקבל כל הזמן ערכים שונים.

נשיג מטרה זו בעזרת הפונקציה **Rnd**, ש"מציגה" כל פעם על המסך מספר מקרי של קובץ-תמונה. ניתן להשוות את תפקוד פונקציה זו עם *"גלגל המזל"*, או עם כל גלגל מסתובב, שבו נמצאים כדורים ממוספרים. כאשר הגלגל נעצר, נופל ממנו כדור כלשהו. מספר כדור זה הוא מספר מקרי, אחד מכל המספרים שנמצאים בגלגל.

בתוכנית שלנו עובדת אותה שיטה: פונקציה **Rnd** "מוציאה" מה"גלגל" מספר אקראי של קובץ ומציבה אותו במשתנה **Pic_Number**:

```
Pic_Number = Int(Rnd * 50)
```

הביטוי **(Rnd * 50)** אומר, שב"גלגל" שלנו יש 50 כדורים (50 הקבצים הממוספרים מ-**0 ... 49**) ומהם נבחר מספר כלשהו. לפני ביטוי זה מוצבת הפונקציה **Int**, שגורמת לסוג המספר הנבחר באופן מקרי להיות סוג שלם - **Integer**.

נפעיל את התוכנית שלנו ונשים לב לסדר הופעת התמונות הראשונות. אם נפעיל אותה מחדש, נראה שהתמונות מופיעות באותו סדר. גם בפעם השלישית והרביעית נשמר הסדר! משחק כזה הוא משעמם ביותר, שהרי אפשר לזכור את הסדר הזה.

כדי למנוע חזרה כזו, עדיף לסובב את "הגלגל" שלנו לפני כל הצגת מספר מקרי, במטרה לערבב את המספרים - "כדורים". ניתן לבצע זאת באמצעות הפונקציה **Randomize** (מהמילה **Random** - אקראי, מקרי) בתחילת הפרוצדורה:

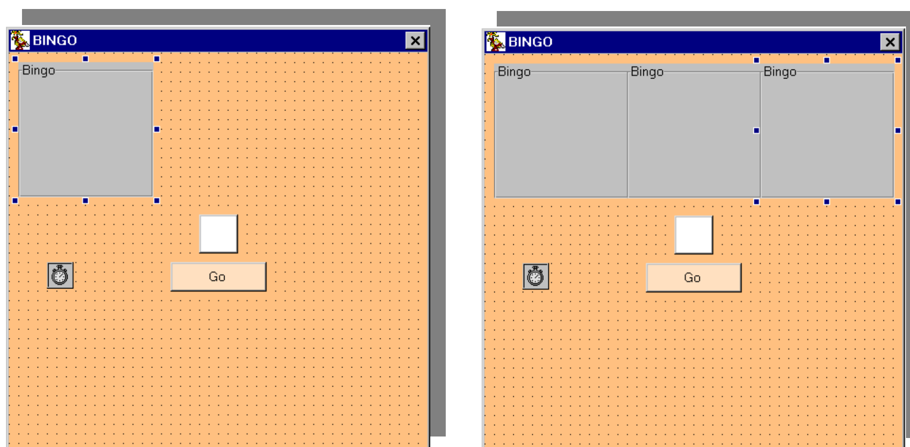
```
Private Sub Timer1_Timer()
   Randomize
   Pic_Number = Int(Rnd * 50)
   cmd_Show.Picture = LoadPicture(Pic_Number)
End Sub
```

כעת "הגלגל" שלנו יסתובב כל שנייה, עם כל "תקתוק" חדש של הטיימר. לאחר הערבוב - **Randomize**, מושלך מ"הגלגל" מספר אקראי, אחד מ-50 המספרים שלנו **(Rnd * 50)**. המספר הופך לשלם ומוצב במשתנה **Pic_Number**. הפרוצדורה **LoadPicture** טוענת קובץ עם מספר זה כתמונה (**Picture**) ללחצן **cmd_Show**.

נפעיל את התוכנית שוב. מעתה, בכל פעם שנפעיל את התוכנית מחדש, סדר הופעת התמונות ישתנה!

"לוחות" ו"כרטיסי" משחק

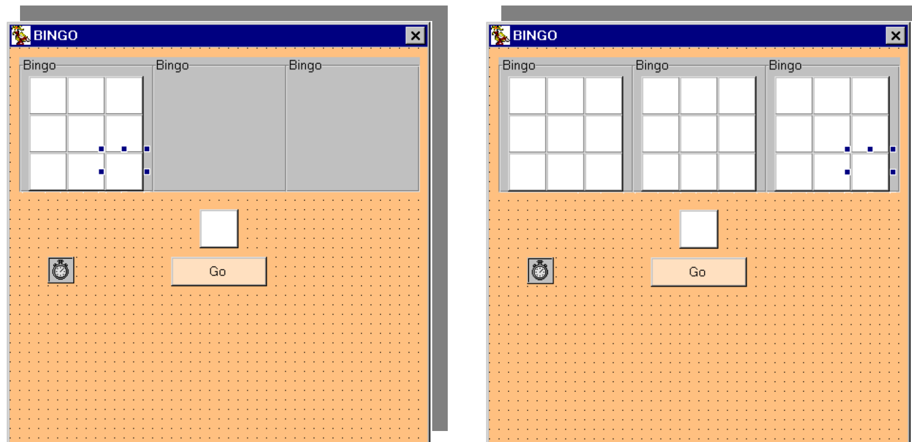
1. נוסיף לטופס מסגרת (**Frame1**) עם הגדלים: **Height - 1700**, **Width - 1700**.
2. בשורת המאפיין **Caption** נכתוב **Bingo**. זה יהיה ה"לוח" הראשון שעליו משחק המחשב.
3. נקרא ללוח-פריים זה **fra_Bingo**.
4. בעזרת הפקודה **Paste**, **Copy** ניצור מערך משולשה לוחות כאלה: **fra_Bingo(0) ... fra_Bingo(2)**.



5. נצייר בתוך הלוח הראשון "כרטיס" - לחצן פקודה בגודל הלחצן **cmd_Show**.
6. נקבע ללחצן את המאפיינים:

- **Name - cmd_Bingo**
- **Style - Graphical**
- **BackColor - לבן**
- **Caption - נעדר**

7. נעתיק (**Copy**) לחצן זה ונסמן באמצעות העכבר את הלוח-פריים הראשון.
8. בעזרת הפקודה **Paste** ניצור מערך מ-9 הלחצנים-כרטיסים בתוך הלוח-פריים.
יש להקפיד שבזמן ביצוע הפקודה **Paste**, לוח-פריים יהיה מסומן באמצעות העכבר. אם לא נעשה זאת, חלק ממערך הלחצנים-כרטיסים יימצא מחוץ ללוח ויגרום ל"באגים" בתפקוד התוכנית.

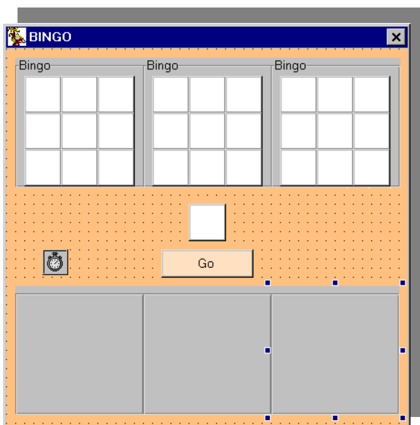


9. נבדוק את מיקום הכרטיסים על הלוח: אם המערך נוצר נכון, התוצאה שתתקבל היא תשעה לחצנים-כרטיסים עם שמות מ-**cmd_Bingo(0)** עד **cmd_Bingo(8)**. אי אפשר להוריד את הלחצנים מהפריים. אם נגזור את הפריים בעזרת הפקודה **Cut**, כל הכרטיסים ייעלמו יחד איתו.

10. תכונה זו של המסגרת-פריים שלנו תשמש אותנו בצורה נוחה מאוד בהמשך עבודת התוכנית. מספיק יהיה רק להגדיר את מאפיין הפריים **Visible** כ-**False**, וכל הכרטיסים הנמצאים עליו, ייהפכו לבלתי נראים.

11. באופן זה בדיוק נמקם בכל שאר הלוחות תשעה כרטיסים. אם מערך הלחצנים נוצר נכון, אז מספר (אינדקס) הכרטיס האחרון יהיה 26, מכיון שבמשחק משתתפים 27 כרטיסים בסך הכל.

12. עתה נותר ליצור מערך משלושה לוחות-פריימים לשחקנים עם השמות **fra_Player(0) ... fra_Player(2)**, אך נשאיר את המאפיין **Caption** ריק.

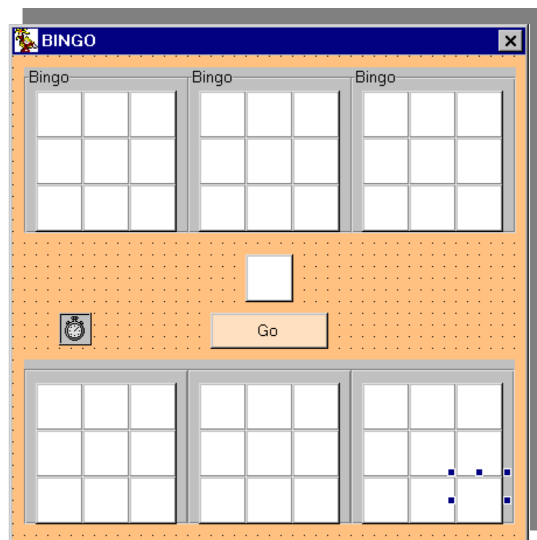


13. לתוך הלוח הראשון נוסיף כרטיס-לחצן בשם **cmd_Player**. ללחצן זה יהיו אותם מאפיינים שיש ללחצן **cmd_Bingo**.

14. ניצור מערך מלחצני **cmd_Player(0)... cmd_Player(26)**, ונמקמם בתוך לוחות הפריים.

אם עיצוב הטופס בוצע נכון, הטופס צריך להיראות כמו בתמונה.

כעת, אפשר לצבוע את הלוחות בצבעים שונים ולשפר את מראה המשחק.



נמלא את הכרטיסים בתמונות

לפי כללי משחק "בינגו" המקורי, כל שחקן מקבל כרטיסים עם תמונות. גם עלינו מוטלת המשימה למלא בתמונות את הכרטיסים שממוקמים על הלוחות, ולעשות זאת כך שהתמונות לא תחזורנה באותו סדר בכל הפעלה מחדש של המשחק. נשתמש שוב בשיטה המוכרת לנו - בחירה מקרית (רנדומיזציה).

1. עלינו להגדיר בחלק **General** עוד מספר משתנים:

- **Dim** i **As integer** משתנה-מונה הלולאה
- **Dim** Pic_Bingo **As Integer** מספר התמונה של בינגו
- **Dim** Pic_Player **As Integer** מספר התמונה של השחקן

2. בהפעלת התוכנית מתבצעת שגרת טעינת הטופס **Private Sub Form_Load()**. נכתוב:

```
Private Sub Form_Load()
 For i = 0 To 26
   Randomize
   Pic_Bingo = Int(Rnd * 50)
   cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
   Pic_Player = Int(Rnd * 50)
   cmd_Player(i).Picture = LoadPicture(Pic_Player)
 Next
End Sub
```

3. נפעיל את התוכנית שלנו. על המסך יופיע הטופס עם הכרטיסים המלאים בתמונות שונות. עם הפעלת התוכנית מחדש, התמונות שעל הכרטיסים צריכות להשתנות. מספר תמונות מופיעות פעמיים או יותר, מכיון שהפונקציה **Rnd** יכולה בהחלט להשליך אותו מספר כמה פעמים.

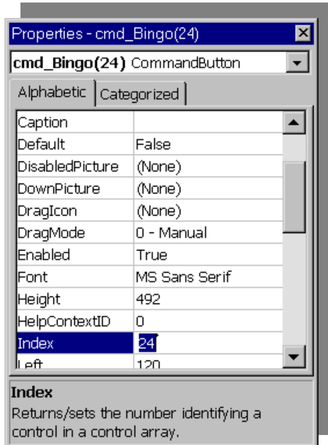


קצת תיאוריה לגיניים יותר

השתמשנו כבר בלולאות ובמערכים, כאשר כתבנו את התוכנית הקודמת. כעת נדון באופן פעולתם.

קודם כל, מה בתוכנית שלנו מהווה מערך?

יצרנו **27** לחצני **cmd_Bingo** ו-**27** לחצני **cmd_Player**. לכל הלחצנים אותו שם - הם נבדלים רק במספר הרשום בסוגריים, למשל **cmd_Bingo(5)**, או **cmd_Player(21)**. מספר זה נקרא אינדקס מרכיב המערך (**Index**). בחלון המאפיינים אפשר לראות את האינדקס בחלק העליון, בשם המרכיב, וגם בשורת המאפיין המיוחדת שנקראת - **Index**.



Properties - cmd_Bingo(24)

cmd_Bingo(24) CommandButton

Alphabetic | Categorized

Caption	
Default	False
DisabledPicture	(None)
DownPicture	(None)
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif
Height	492
HelpContextID	0
Index	24
Left	120

Index
Returns/sets the number identifying a control in a control array.

כדי לבצע פקודה זהה לכל מרכיבי המערך, למשל, למלא בתמונות את לחצני-כרטיסי **cmd_Bingo**, אין צורך לחזור על הפקודה **27** פעמים. מספיק לתכנת את הלולאה, שתחזור על עצמה **27** פעמים:

```
For i = 0 To 26
```

המשתנה **i** שבביטוי, משנה את ערכו מ-**0** עד **26** (**27** פעמים). במשך התוכנית יש למלא בתמונות את הכרטיס הראשון, אחר את השני, ואחרי את כל השאר.

במשך התוכנית שלנו אנו צריכים למלא בתמונות את הכרטיס הראשון, אחרי את השני, ואחרי את כל השאר.

הלאה, אנו צריכים שבכל חזרה חדשה (איטרציה) של הלולאה אינדקס הלחצן יגדל ב-**1**. כך, שבחזרה הראשונה הוא שווה ל-**0**, בשנייה ל-**1**, בשלישית - **2** וכו'... תופעה זו מושגת באמצעות שיטה פשוטה מאוד: אנו מחליפים את האינדקס במשתנה **i**, אשר גדל אוטומטית ב-**1** בכל חזרה של הלולאה. למטרה זו עלינו להחליף בשם הלחצן את האינדקס במשתנה **i**, כלומר, לכתוב **cmd_Bingo(i)**.

מעכשיו בחזרה הראשונה של הלולאה מתבצעת הפקודה:

```
cmd_Bingo(0).Picture = LoadPicture(Pic_Bingo)
```

בחזרה השנייה של הלולאה מתבצעת הפקודה:

```
cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
```

וכך עד לסוף הלולאה, כשכל **27** הכרטיסים יהיו מלאים בתמונות.

"בינגו" נכנס למשחק

הטופס שלנו מוכן לתכנות המשחק. כעת נכתוב את הפרוצדורה שבה "בינגו", כלומר המחשב, משחק אוטומטית. אם תמונה אחת או מספר תמונות, שעל כרטיסי המחשב, זהות לתמונה שעל "המסך", תמונות אלו צריכות להיעלם.

אך כיצד ניתן להשוות את התמונה על "המסך" עם התמונות שעל הכרטיסים?

בין המאפיינים של מרכיבי Visual Basic קיים אחד שיכול להועיל לנו מאוד. מאפיין זה נקרא **Tag** - תווית. על "תווית" זו אפשר לכתוב מספר כלשהו או אות, מילה או כמה מילים. יהיה לנו נוח יותר להציב במאפיין זה מספר, הזהה למספר קובץ התמונה. יש לבצע זאת ברגע טעינת התמונות לכרטיסים. לשם כך, צריך להוסיף לפרוצדורת טעינת הטופס שתי פקודות: אחת - לכרטיסי בינגו, והשנייה - לכרטיסי השחקן.

```
Private Sub Form_Load()
        For i = 0 To 26
          Randomize
          Pic_Bingo = Int(Rnd * 50)
          cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
          cmd_Bingo(i).Tag = Pic_Bingo
          Pic_Player = Int(Rnd * 50)
          cmd_Player(i).Picture = LoadPicture(Pic_Player)
          cmd_Player(i).Tag = Pic_Player
        Next
End Sub
```

כאן אנו כאילו מדביקים על כל אחד מהכרטיסים את התווית (**Tag**) עם מספר הקובץ של התמונה שנמצאת בכרטיס. התווית (**Tag**) נסתרת עבורנו, אך היא מראה לתוכנית את מספר התמונה שעל הכרטיס.

כעת הטיימר צריך לבדוק, האם לפחות אחד מ-**27** המספרים שעל תוויות הכרטיסים העליונים מתאים למספר קובץ התמונה שמופיע על "המסך". אם כן, הכרטיס הזה (או מספר כרטיסים כאלה) ייעלם.

```
Private Sub Timer1_Timer()
    Randomize
    Pic_Number = Int(Rnd * 50)
    cmd_Show.Picture = LoadPicture(Pic_Number)
       For i = 0 To 26
            If cmd_Bingo(i).Tag = Pic_Number Then
                cmd_Bingo(i).Visible = False
                End If
       Next
End Sub
```

נבדוק את התוכנית. אם נחכה בסבלנות, לאחר זמן מה כל הכרטיסים שעל הלוחות-פריימים העליונים ייעלמו. כדי לא לחכות זמן רב, אפשר להקטין בינתיים את אינטרוול הטיימר עד **300-500** אלפיות השנייה.

"בינגו" סופר נקודות

1. כרטיס שנעלם הוא נקודה לזכות "בינגו". לחישוב הנקודות נגדיר משתנה:

```
Dim Bingo_Score As Integer,
```

משתנה זה יגדל בכל פעם שמספר תמונת הכרטיס זהה למספר התמונה שעל "המסך":

```
If cmd_Bingo(i).Tag = Pic_Number Then
   cmd_Bingo(i).Visible = False
   Bingo_Score = Bingo_Score + 1
End If
```

2. כאשר מספר הנקודות שווה **ל-27** המשחק ייעצר ובכותרת הטופס תופיע הודעה ש"בינגו" ניצח. בפרוצדורת הטיימר יש לכלול עכשיו את הפקודות הבאות:

```
Private Sub Timer1_Timer()
Randomize
Pic_Number = Int(Rnd * 50)
cmd_Show.Picture = LoadPicture(Pic_Number)
  For i = 0 To 26
    If cmd_Bingo(i).Tag = Pic_Number Then
      cmd_Bingo(i).Visible = False
      cmd_Bingo(i).Tag = 100
      Bingo_Score = Bingo_Score + 1
    End If
  Next
    If Bingo_Score = 27 Then
      Timer1.Enabled = False
      Form1.Caption = "Bingo won!"
    End If
End Sub
```

הביטוי **cmd_Bingo(i).Tag = 100** נראה לנו מוזר, לא? מדוע אנו צריכים לתת לתוויות הכרטיסים שנעלמו ערך השווה **ל-100**? ואם כבר, למה לא **500** או **1000**?

נבדוק למה, נסיר פקודה זו מהפרוצדורה ונפעיל את התוכנית - הטיימר ייעצר הרבה לפני שייעלמו כל הכרטיסים העליונים. מדוע?

מתברר של- Visual Basic לא חשוב האם הלחצן-כרטיס גלוי, או לא. אפילו אם הוא נעלם מהמסך, הוא לא נעלם מזיכרון המחשב והמחשב ממשיך להתחשב בו. ובכן, אם על הלחצן-"מסך" תופיע תמונה שהיתה בכרטיס שנעלם, המשתנה **Bingo_Score** יגדל באחד גם כך. זאת אומרת, שחישוב הנקודות עד **27** ייגמר לפני שתיעלמנה כל התמונות. אם נציב במאפיין **Tag** של הכרטיס מספר שאינו קיים בין מספרי התמונות, דבר זה לא יקרה.

נשחק עם "בינגו"

כאשר השחקן רואה על "המסך" את התמונה הזהה לתמונה שעל הכרטיס שלו, הוא צריך ללחוץ על כרטיס זה. הכרטיס נעלם ואיתו נעלמים כל שאר הכרטיסים שעליהם תמונה זו קיימת, ולמונה נוסף מספר נקודות השווה למספר הכרטיסים שנעלמו.

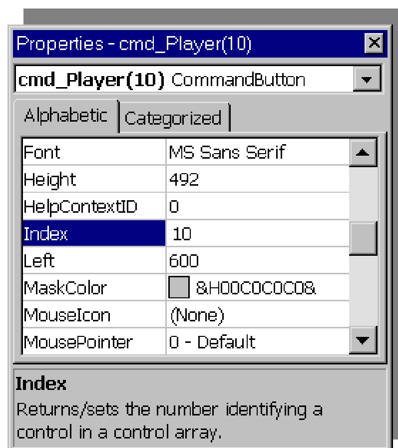
1. נגדיר משתנה-מונה נקודות לשחקן:

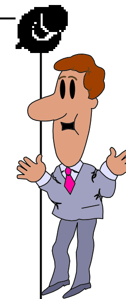
```
Dim Player_Score As Integer
```

2. נכתוב פרוצדורה ללחצנים התחתונים:

```
Private Sub cmd_Player_Click(Index As Integer)
If cmd_Player(Index).Tag = Pic_Number Then
   cmd_Player(Index).Visible = False
End If
End Sub
```

נשים לב לביטוי בסוגריים שבכותרת הפרוצדורה **(Index As Integer)** וגם למשתנה **Index**, המשמש כמונה מרכיבי המערך **cmd_Player(Index)**.





קודם כל, נזכיר שוב מהו אינדקס. כאשר אנו יוצרים את מערך המרכיבים, Visual Basic מציב בהם מספרים באופן אוטומטי. את המספרים האלה ניתן לראות בחלון המאפיינים. מספיק לסמן בעכבר מרכיב כלשהו, ובחלק העליון של חלון המאפיינים יופיע שם המרכיב ומספרו בסוגריים. ניתן גם לראות מספר זה בשורת המאפיין שנקרא **Index**.



נניח שעל "המסך" מופיעה תמונה שנמצאת גם על אחד מכרטיסי השחקן.

הלחצן, שעליו השחקן צריך ללחוץ - **cmd_Player** - עם המספר (אינדקס) **15**

השחקן צריך ללחוץ על לחצן זה, כדי להסיר את הכרטיס. אבל איך נודיע לתכנית איזה לחצן נלחץ? הרי מדובר במערך לחצנים, שיש להם אותו שם - **cmd_ Player**...

Visual Basic מזהה את מרכיבי המערך לפי המספרים-אינדקסים. נניח שיש להסיר את כרטיס מספר **15**. נכתוב את הפקודה הבאה:

```
cmd_Player(15).Visible = False,
```

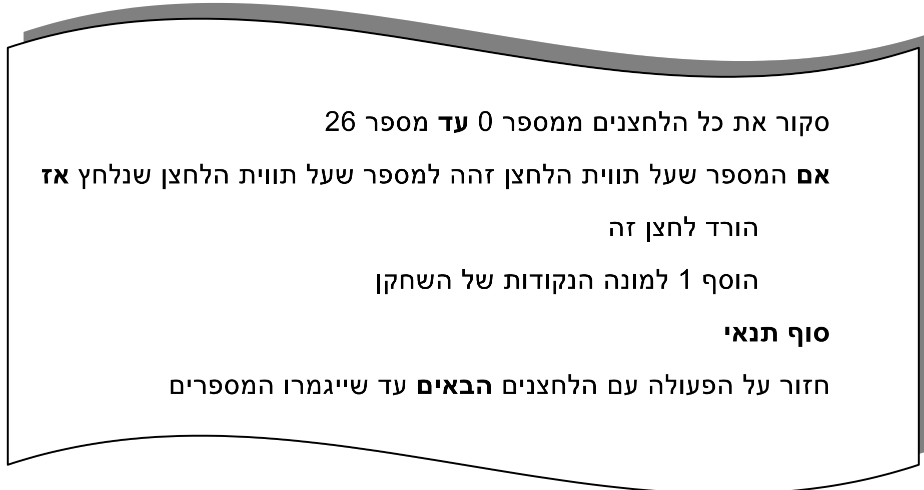
הלחצן יהפוך ללא גלוי.

אך במשחק אמיתי אנו צריכים שכל אחד מהלחצנים יוכל להיות לא גלוי. כיצד נעשה זאת? נחליף את המספר המוגדר למשתנה **Index**, שדרך אגב כבר קיים בכותרת הפרוצדורה בצורת **Integer**, ואיננו חייבים להגדיר אותו בנפרד. מעכשיו, אפשר למחוק כל לחצן-כרטיס באמצעות לחיצת עכבר:

```
Private Sub cmd_Player_Click(Index As Integer)
   If cmd_Player(Index).Tag = Pic_Number Then
      cmd_Player(Index).Visible = False
   End If
End Sub
```

נבדוק את עבודת הפרוצדורה שלנו. כל אחד מהלחצנים נמחק בקלות, אם ה-**Tag** שלו תואם למספר קובץ התמונה שמופיעה על הלחצן-"מסך". בכל זאת יש בעיה אחת רצינית: אם היו לנו שני כרטיסים או יותר רק אחד מהם נמחק: הכרטיס שעליו לחצנו בעכבר. כיצד נמחק את שאר הכרטיסים?

נארגן את בדיקת המערך כדי לגלות תמונות זהות, כלומר נכתוב את הלולאה כך :



סקור את כל הלחצנים ממספר 0 **עד** מספר 26

אם המספר שעל תווית הלחצן זהה למספר שעל תווית הלחצן שנלחץ **אז**

הורד לחצן זה

הוסף 1 למונה הנקודות של השחקן

סוף תנאי

חזור על הפעולה עם הלחצנים **הבאים** עד שייגמרו המספרים

נתרגם את הלולאה הזאת לשפת Visual Basic :

```
For i = 0 To 26
 If cmd_Player(i).Tag = cmd_Player(Index).Tag Then
   cmd_Player(i).Visible = False
   Player_Score = Player_Score + 1
 End If
Next
```

מעכשיו, כל התמונות הזהות תימחקנה וחישוב הנקודות יהיה תלוי במהירות תגובת השחקן, כלומר, האם הספיק ללחוץ על הלחצן-כרטיס שלו.

סיכומים...

נשאר לנו להוסיף לפרוצדורת חישוב הנקודות את פקודת עצירת הטיימר ברגע שמספר הכרטיסים שנמחקו מהלוחות השחקן יגיע ל -**27**:

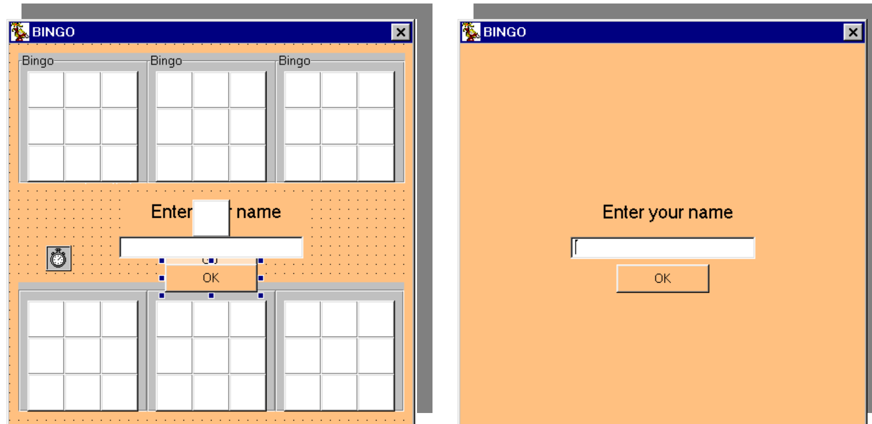
```
Private Sub Timer1_Timer()
Randomize
Pic_Number = Int(Rnd * 50)
cmd_Show.Picture = LoadPicture(Pic_Number)
  For i = 0 To 26
    If cmd_Bingo(i).Tag = Pic_Number Then
      cmd_Bingo(i).Visible = False
      cmd_Bingo(i).Tag = 100
      Bingo_Score = Bingo_Score + 1
    End If
  Next
    If Bingo_Score = 27 Then
      Timer1.Enabled = False
      Form1.Caption = "Bingo won!"
    End If
If Player_Score = 27 Then
      Timer1.Enabled = False
    End If
End Sub
```

זהו זה. התוכנית נכתבה, אפשר לשחק! אך מי שרוצה שהמשחק שלו ייראה "אמיתי", צריך להצטייד בסבלנות ולכתוב עוד כמה תוספות.

יצירת ממשק המשחק

1. נסמן בעכבר את כל ששת הלוחות-פריימים, וגם את לחצני **cmd_Show** ו-**cmd_Play**, ונקבע לכולם **בו-זמנית** את המאפיין **False - Visible**.
2. נוסיף מעל הלחצנים תווית - **Label** ונכתוב בשורת **Caption** של התווית "**Enter your name**". נמרכז את הכיתוב (מאפיין **Center - Alignment**), נבחר גודל גופן (מאפיין **Font**) ונצבע את התווית בצבע כלשהו (מאפיין - **BackColor**).
3. מתחת לתווית נוסיף תיבת טקסט ונסיר את הכיתוב מהמאפיין **Text** שלה.
4. בחלק התחתון נמקם לחצן פקודה עם הכיתוב **OK** ונקרא לו בשם **cmd_OK**.

גם אחרי הפעלת המשחק, הטופס בחלון העיצוב צריך להיראות כמו בתמונות:

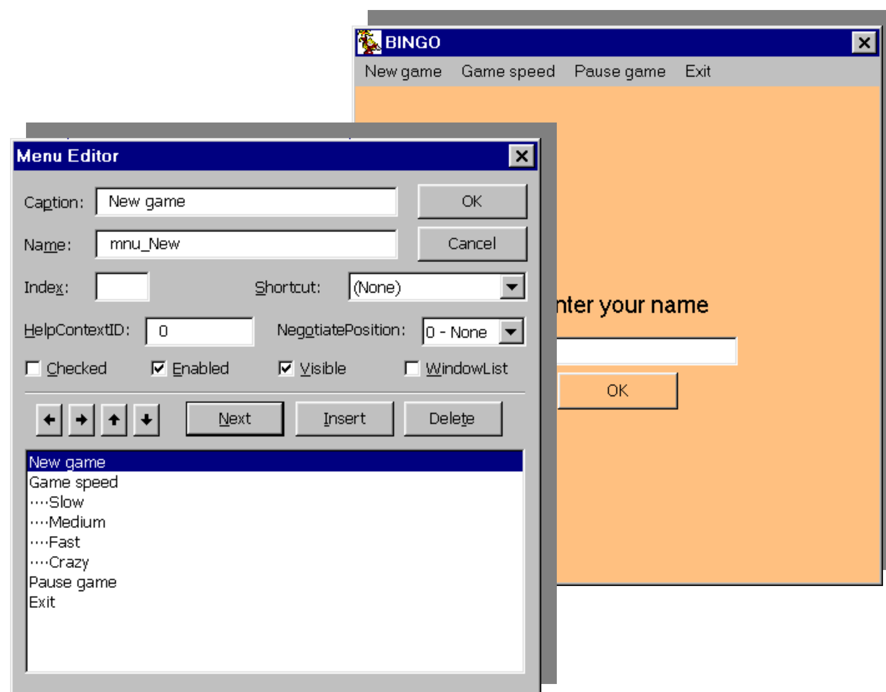


5. נכתוב ללחצן **cmd_OK** את הפרוצדורה שמכניסה לכותרת הלוחות-פריימים את שם השחקן שהוא מקליד לתיבת הטקסט **Text1**, ואחר מראה על הטופס את כל מרכיבי הממשק שהיו נסתרים. נשים לב שיחד עם הלוחות-פריימים גם הלחצנים-כרטיסים הפכו גלויים:

```
Private Sub cmd_OK_Click()
  For i = 0 To 2
    fra_Player(i).Caption = Text1.Text
    fra_Player(i).Visible = True
    fra_Bingo(i).Visible = True
  Next
  cmd_Show.Visible = True
  cmd_Play.Visible = True
  Label1.Visible = False
  Text1.Visible = False
  cmd_OK.Visible = False
End Sub
```

יצירת התפריט

אנו כבר מכירים את שיטת יצירת התפריט - עסקנו בה בשני הפרויקטים הראשונים. בפרק הזה נדגים כיצד צריך להיראות תפריט המשחק שלנו על ידי תמונות וטבלה בלבד.



Caption - כיתוב		Name - שם
New game		**mnu_New**
Game speed		**mnu_Speed**
	Slow	**itm_Slow**
	Medium	**itm_Medium**
	Fast	**itm_Fast**
	Crazy	**itm_Crazy**
Pause game		**mnu_Pause**
Exit		**mnu_Exit**

בחירת קצב המשחק

המהירות שהטיימר מראה תמונות על "מסך" תלויה בבחירת קצב המשחק. את פרוצדורות הבחירה נרשום בפריטי התפריט **Game speed**. בנוסף, כדי שאפשר יהיה לסמן (**CheCked**) את הפריטים הנבחרים, נוסיף לכל פרוצדורה את הפקודות הבאות:

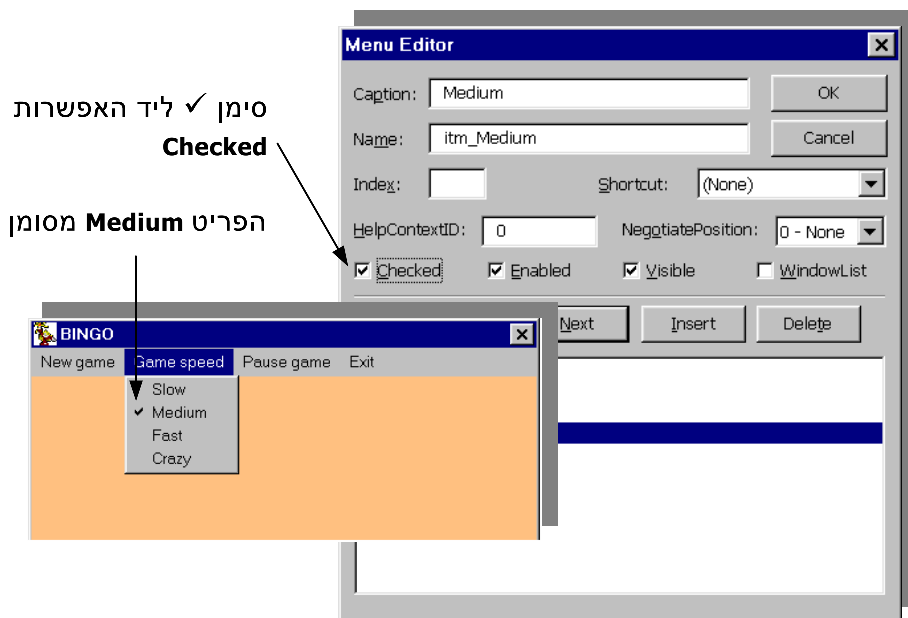
```
Private Sub itm_Slow_Click()
    Timer1.Interval = 3000
    itm_Slow.Checked = True
    itm_Medium.Checked = False
    itm_Fast.Checked = False
    itm_Crazy.Checked = False
End Sub

Private Sub itm_Medium_Click()
    Timer1.Interval = 2000
    itm_Slow.Checked = False
    itm_Medium.Checked = True
    itm_Fast.Checked = False
    itm_Crazy.Checked = False
End Sub

Private Sub itm_Fast_Click()
    Timer1.Interval = 1000
    itm_Easy.Checked = False
    itm_Medium.Checked = False
    itm_Fast.Checked = True
    itm_Crazy.Checked = False
End Sub

Private Sub itm_Crazy_Click()
    Timer1.Interval = 10
    itm_Easy.Checked = False
    itm_Medium.Checked = False
    itm_Fast.Checked = False
    itm_Crazy.Checked = True
End Sub
```

בחלון המאפיינים נקבע לטיימר **Interval - 2000**. קצב זה של המשחק יהיה ברירת המחדל (**Default**), במקרה והשחקן לא ירצה לבחור קצב משחק אחר. לכן לפריט **Medium** של התפריט (שבו קצב המשחק הוא אינטרוול 2000 אלפיות השנייה) נסמן בתפריט את האופציה **Checked**. מעכשיו, בעת הכניסה לתפריט, פריט זה יהיה מסומן על ידי "✓" עד שהשחקן יבחר פריט אחר.



יציאה מהמשחק

את הפרוצדורה הבאה אנו מכירים כבר מהפרויקטים הקודמים:

```
Private Sub mnu_Exit_Click()
   End
End Sub
```

הפסקת המשחק

את פרוצדורת הפסקת המשחק נכתוב לתפריט **Pause game**:

```
Private Sub mnu_Pause_Click()
   If Timer1.Enabled = True Then
      Timer1.Enabled = False
      mnu_Pause.Caption = "Resume game"
   Else
      Timer1.Enabled = True
      mnu_Pause.Caption = "Pause game"
   End If
End Sub
```

קצת תיאוריה

עד עתה השתמשנו בתנאי פשוט "**אם - אז...**" (**If - Then...**), כעת התנאי הפך מורכב יותר: הוספנו לו את המילה **Else**, כלומר - **אחרת**. אם נתרגם את הפרוצדורה שלנו לשפה רגילה, נגיד משהו כזה:

אם הטיימר עובד, **אז** (**If** הטיימר עובד **Then**)

כבה את הטיימר

כתוב לתפריט **Pause** את הכותרת "המשך משחק"

אחרת (**Else**)

הפעל את הטיימר

כתוב לתפריט **Pause** את הכותרת "הפסק משחק"

סוף התנאי "**אם**" (**End If**)



חשוב להבין למה מתיחס **Else** - "אחרת". לפעמים אי-הבנה מסוג זה גורמת לטעויות בתכנות. צריך לזכור ש-**Else** הוא **תנאי שהפוך לחלוטין לתנאי שעומד לאחר המילה IF**.

במקרה שלנו **Else** מבטא את העובדה שהטיימר כבוי, כלומר, תנאי זה יכול לעבוד רק לאחר שכיבינו את הטיימר, אחר, לחיצה בעכבר על תפריט זה תביא להרצת הטיימר ותחזיר את ה-**Caption** למצבו ההתחלתי - **Pause game**.

פרוצדורת ההחלפה הפשוטה הזו מאפשרת לעצור את המשחק (ליתר דיוק - את הטיימר), ואחר לחדש אותו. הכותרת משתנה תוך כדי פעולות אלו: **Pause game** אם הטיימר עובד, או **Resume game** אם הטיימר כבה.

כש"בינגו" מנצח...

עכשיו ננסה לשחק במשחק שלנו. אם נתחיל בקצב משחק מהיר או אפילו בינוני - סבירות הניצחון תהיה זעירה! הרי "בינגו" אף פעם אינו טועה, וכל תמונה ש"נפספס" תקרב אותנו להפסד בלתי-נמנע. לכן, כדאי לשחק את המשחק הראשון בקצב האיטי, ואם בכל זאת ההפסד הוא בלתי נמנע, ניתן ל"בינגו" לנצח ומהר! נבחר את הפריט **Crazy** ובתוך מספר שניות הלוחות העליונים יתרוקנו לחלוטין! משחק חדש...

1. נפתח את פרוצדורת התפריט **New Game**:

```
Private Sub mnu_New_Click()
```

2. "נחביא" את כל הלוחות:

```
For i = 0 To 2
   fra_Bingo(i).Visible = False
   fra_Player(i).Visible = False
 Next
```

3. "נחביא" את לחצן ה"מסך" ואת לחצן **Go**:

```
 cmd_Show.Visible = False
 cmd_Play.Visible = False
```

4. נחזיר את הכיתוב הקודם לכותרת הטופס:

```
Form1.Caption = "BINGO"
```

5. נחזיר את התווית עם הכיתוב, את תיבת הטקסט ואת לחצן **OK**:

```
Label1.Visible = True
Text1.Visible = True
cmd_OK.Visible = True
```

6. נמחק מתיבת הטקסט את הטקסט הישן:

```
Text1.Text = ""
```

7. נאפס את המונים (של השחקן ושל "בינגו"):

```
 My_Score = 0
 Bingo_Score = 0
```

8. נכניס את התמונות לכרטיסי הלוחות העליונים והתחתונים (בדיוק כפי שעשינו בפרוצדורת טעינת הטופס):

```
For i = 0 To 26
        cmd_Player(i).Visible = True
        cmd_Bingo(i).Visible = True
        Randomize
        Pic_Bingo = (Rnd * 49)
        cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
        cmd_Bingo(i).Tag = Pic_Bingo
        Randomize
        Pic_Player = (Rnd * 49)
        md_Player(i).Picture = LoadPicture(Pic_Player)
        cmd_Player(i).Tag = Pic_Player
Next
```

9. נסיר את התמונה הישנה מלחצן ה"מסך":

```
cmd_Show.Picture = LoadPicture()
```

10. נכבה את הטיימר (פקודה זו אינה הכרחית, מכיון שלפי היגיון המשחק הטיימר נכבה לאחר ניצחון של אחד הצדדים. בכל זאת כדאי להוסיפה, כי ייתכן שהשחקן ילחץ על **New Game** לפני שהמשחק ייגמר).

```
Timer1.Enabled = False
```

נאחל הצלחה וניצחון מהיר על "בינגו".

לאלה שרוצים לבדוק את התוכנית שוב, להלן הטקסט המלא של התוכנית.

טקסט התוכנית המלא

```
Dim Pic_Number As Integer
Dim i As Integer
Dim Pic_Bingo As Integer
Dim Pic_Player As Integer
Dim Bingo_Score As Integer
Dim Player_Score As Integer

Private Sub cmd_OK_Click()
  For i = 0 To 2
    fra_Player(i).Caption = Text1.Text
    fra_Player(i).Visible = True
    fra_Bingo(i).Visible = True
  Next
  cmd_Show.Visible = True
  cmd_Play.Visible = True
  Label1.Visible = False
  Text1.Visible = False
  cmd_OK.Visible = False
End Sub

Private Sub cmd_Play_Click()
    Timer1.Enabled = True
End Sub

Private Sub cmd_Player_Click(Index As Integer)
  If cmd_Player(Index).Tag = Pic_Number Then
     cmd_Player(Index).Visible = False
         For i = 0 To 26
           If cmd_Player(i).Tag = cmd_Player(Index).Tag Then
                 cmd_Player(i).Visible = False
                 Player_Score = Player_Score + 1
           End If
          Next
    End If
End Sub

Private Sub Form_Load()
 For i = 0 To 26
  Randomize
  Pic_Bingo = Int(Rnd * 49)
```

```
  cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
  cmd_Bingo(i).Tag = Pic_Bingo
  Pic_Player = Int(Rnd * 49)
  cmd_Player(i).Picture = LoadPicture(Pic_Player)
  cmd_Player(i).Tag = Pic_Player
 Next
End Sub

Private Sub itm_Crazy_Click()
    Timer1.Interval = 10
    itm_Slow.Checked = False
    itm_Medium.Checked = False
    itm_Fast.Checked = False
    itm_Crazy.Checked = True
End Sub

Private Sub itm_Fast_Click()
    Timer1.Interval = 1000
    itm_Slow.Checked = False
    itm_Medium.Checked = False
    itm_Fast.Checked = True
    itm_Crazy.Checked = False
End Sub

Private Sub itm_Medium_Click()
    Timer1.Interval = 2000
    itm_Slow.Checked = False
    itm_Medium.Checked = True
    itm_Fast.Checked = False
    itm_Crazy.Checked = False
End Sub

Private Sub itm_Slow_Click()
    Timer1.Interval = 3000
    itm_Slow.Checked = True
    itm_Medium.Checked = False
    itm_Fast.Checked = False
    itm_Crazy.Checked = False
End Sub

Private Sub mnu_Exit_Click()
    End
End Sub
```

```
Private Sub mnu_New_Click()
    For i = 0 To 2
        fra_Bingo(i).Visible = False
        fra_Player(i).Visible = False
    Next
  Form1.Caption = "BINGO"
  cmd_OK.Visible = True
  Text1.Visible = True
  Text1.Text = ""
  Label1.Visible = True
  cmd_Show.Visible = False
  cmd_Play.Visible = False
  My_Score = 0
  Bingo_Score = 0
    For i = 0 To 26
       cmd_Player(i).Visible = True
       cmd_Bingo(i).Visible = True
       Randomize
       Pic_Bingo = (Rnd * 49)
       cmd_Bingo(i).Picture = LoadPicture(Pic_Bingo)
       cmd_Bingo(i).Tag = Pic_Bingo
       Randomize
       Pic_Player = (Rnd * 49)
       cmd_Player(i).Picture = LoadPicture(Pic_Player)
       cmd_Player(i).Tag = Pic_Player
    Next
    cmd_Show.Picture = LoadPicture()
    Timer1.Enabled = False
End Sub
```

```
Private Sub mnu_Pause_Click()
    If Timer1.Enabled = True Then
        Timer1.Enabled = False
        mnu_Pause.Caption = "Resume game"
    Else
        Timer1.Enabled = True
        mnu_Pause.Caption = "Pause game"
    End If
End Sub
```

```
Private Sub Timer1_Timer()
    Randomize
    Pic_Number = Int(Rnd * 49)
    cmd_Show.Picture = LoadPicture(Pic_Number)
      For i = 0 To 26
          If cmd_Bingo(i).Tag = Pic_Number Then
              cmd_Bingo(i).Visible = False
              cmd_Bingo(i).Tag = 100
              Bingo_Score = Bingo_Score + 1
          End If
      Next
      If Bingo_Score = 27 Then
          Timer1.Enabled = False
          Form1.Caption = "Bingo won!"
      End If
      If Player_Score = 27 Then
        Timer1.Enabled = False
        Form1.Caption = Text1.Text + " won!"
      End If
End Sub
```

5 משחק "מלחמה"

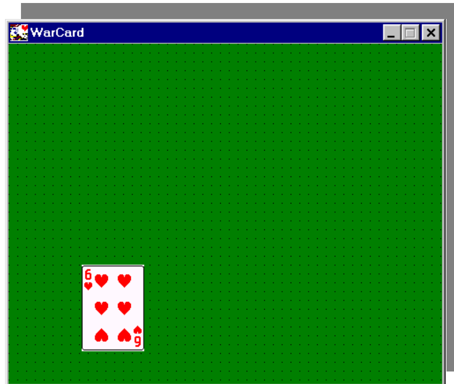
כללי המשחק

"מלחמה" הוא המשחק הפשוט בין כל משחקי הקלפים. חפיסת הקלפים נטרפת ומחולקת חצי-חצי בין שני השחקנים. כל שחקן מוציא מהחפיסה שלו קלף כלשהו. השחקן שערך הקלף שלו גבוה יותר הוא המנצח, וקלף השחקן השני עובר אליו. אם שני השחקנים משליכים קלפים שלהם ערך זהה - פורצת "מלחמה". שניהם מניחים על הקלפים האלה מספר קלפים הפוכים (אחד, שניים או שלושה - לפי סיכום מראש). כשהקלף האחרון גלוי, ושוב - השחקן שערך הקלף שלו גבוה יותר הוא המנצח, ולוקח לעצמו את כל הקלפים שהשליך "למלחמה" יריבו. לפעמים "המלחמות" מתרחשות פעמיים ויותר. זה קורה אם הקלפים האחרונים שהוצגו, שווים.

המנצח במשחק הוא השחקן שיצליח למשוך מיריבו את כל הקלפים שלו.

נידור את ממשק המשחק

1. נפתח פרויקט חדש ונגדיר את מידות הטופס: **Width** = **6000**, **Height** = **5000**.
2. לכותרת הטופס נקליד **WarCard - Caption** ונכניס את התמונה **Icon** - קובץ **Big2.ico** מהתיקיה **Icons**, שבתקליטור המצורף. את המאפיין **MaxButton** נקבע **False**.
3. כרקע לטופס נבחר צבע כלשהו (מאפיין **BackColor**).
4. נוסיף לטופס מסגרת-**Image** ונקרא לה בשם **img_Card**.
5. נכניס למסגרת קובץ **1.bmp** (מאפיין **Picture**) מהתיקיה Images\Card Game, שבתקליטור המצורף.



6. נעתיק את המסגרת עם הקלף (פקודה **Copy**) וניצור את מרכיב המערך הבא (פקודה **Paste** - על השאלה על יצירת מערך נענה "**Yes**").

7. עלינו ליצור חפיסת קלפים - מערך מ-**36** מרכיבים:
img_Card(0) ... img_Card(35).
לכל מסגרת צריך להכניס קובץ-קלף אך **לא לפי הסדר**! כלומר, התמונה הבאה יכולה להיות למשל, קובץ **17.bmp**, אחריו - **21.bmp**, אחריו - **6.bmp** וכדומה. הרי בחפיסה האמיתית נמצאים קלפים בסדר אקראי. כדי לא לבלבל את הקבצים ולא לפספס אף תמונה, רצוי לערוך רשימה של כל 36 המספרים, ולסמן את אלה שבהם השתמשנו כבר.

8. צריך להעניק תג (**Tag**) מסוים לכל קלף, ללא קשר לסוגו:

- שש **1**
- שבע **2**
- שמונה **3**
- תשע **4**
- עשר **5**
- נסיך **6**
- מלכה **7**
- מלך **8**
- אס **9**

יש להיזהר בעת הענקת התגים מכיון שהמחשב אינו מבין כלום במשחקי קלפים, ותגי הקלפים (**Tags**) מהווים בשבילו את התכונה היחידה, שלפיה יוכל לזהות את הקלפים ולהשוות ביניהם. אם במקרה נבלבל תגים (**Tags**), המשחק יתבלבל: הקלף שלו ערך נמוך יותר ימשוך את הקלף שלו ערך גבוה יותר.

9. לאחר שכל מרכיבי המערך יתמלאו בתמונות-קלפים, נסמן אותם בעכבר ונקבע בשביל כולם מאפיינים **Left = 1000** ו- **Top = 3000**.

בחלק השמאלי של הטופס נמצאת כעת חפיסה מ-**36** קלפים. לפי כללי המשחק אנו צריכים לחלק אותה לשני חלקים שווים, אך לפני כן, יש לערבב אותם כדי שבכל משחק חדש סדר הקלפים בכל חלק יהיה שונה.

נגיד שהמחשב ישחק בקלפים הנמצאים בחלק השמאלי של הטופס, ולשחקן שייכים הקלפים שנמצאים בחלק הימני. לקלפי המחשב נשתמש במשתנה-מונה **i**, לקלפי השחקן נשתמש במשתנה-מונה **j**. נגדיר את מיקום קלפי השחקן : הקואורדינטות שלהם בטופס - **Left = 4000** ו-**Top = 3000**.

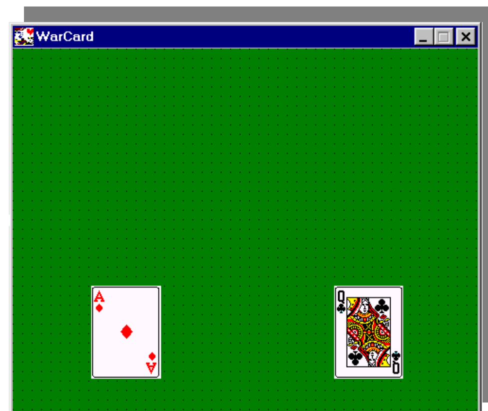
נחלק קלפים...

פרוצדורת החלוקה בין שתי חפיסות של **18** קלפים בכל אחת - אמורה לעבוד בזמן טעינת הטופס (**Form Load**). כיצד נסדיר זאת?

1. נגדיר בחלק **General** שני משתנים-מונים של מספרי הקלפים :

```
Dim i,j As Integer
```

2. כעת עלינו לערבב שוב את החפיסה. ניתן לעשות זאת בעזרת הפקודה המוכרת לנו - **Randomize** ולאחר מכן להוציא מהחפיסה מספר מקרי כלשהו (אינדקס) של הקלף **i = (Rnd * 36)**.



קלפי השחקן - מונה **i** קלפי המחשב - מונה **j**

3. לאחר מכן, יש להבטיח שהקלף עם מספר זה יעבור מהחפיסה השמאלית לימנית :

```
img_Card(i).Left = 4000
```

4. כדי שהפעולה תחזור על עצמה **18** פעמים, נוסיף לחלק **General** משתנה נוסף - מונה קלפי השחקן :

```
Dim Player_Cards As Integer
```

5. ניצור לולאה שבה הפקודות הבאות תחזורנה על עצמן **18** פעמים:

```
Do Until Player_Cards = 18
        Randomize
        i = (Rnd * 36)
        img_Card(i).Left = 4000
        Player_Cards = Player_Cards + 1
 Loop
```

כל הפקודות שבין תחילת (**Do Until**) הלולאה לסופה (**Loop**) תחזורנה על עצמן עד שהמשתנה **Player_Cards** יהיה שווה ל-**18**, כלומר - לשחקן בחפיסה הימנית יהיו **18** קלפים.

אך אם נבדוק את חלוקת הקלפים בפועל (ניתן לעשות את זה בעזרת הפרוצדורה המאוחדת **DragDrop** שעליה נדבר בפרויקט אחר), לשחקן תמיד יהיה פחות מ-**18** קלפים, מדוע?

ייתכן מאוד שהפרוצדורה **i = (Rnd * 36)** "תוציא" יותר מפעם אחת את אותו מספר מקרי. מכיון שהקלף עם המספר (האינדקס) הזה הועבר כבר לחפיסה הימנית, ביצוע הפקודה **img_Card(i).Left = 4000** לא יוביל לשום דבר, אך למונה הקלפים **Player_Cards** יתווסף אחד.

כדי שהדבר לא יקרה, עלינו להכניס את התנאי הבא: ניתן להעביר לחפיסה **הימנית** (**Left** = **4000**) רק את הקלף שנמצא בחפיסה **השמאלית** (**Left** = **1000**):

```
If img_Card(i).Left = 1000 Then
  img_Card(i).Left = 4000
```

אם אין שם קלף עם המספר (האינדקס) הזה, יש לחזור לתחילת הפרוצדורה, לערבב שוב את החפיסה, להוציא קלף מקרי ולבדוק האם אפשר להעביר אותו לחפיסה הימנית. כך נראית הפרוצדורה:

```
Private Sub Form_Load()
  Do Until Player_Cards = 18
1:       Randomize
        i = (Rnd * 35)
        If img_Card(i).Left = 1000 Then
           img_Card(i).Left = 4000
           Player_Cards = Player_Cards + 1
        Else
           GoTo 1
        End If
  Loop
 End Sub
```

ההעברה לתחילת הפרוצדורה מתבצעת בעזרת הפקודה **GoTo**, אחריה יש להפנות למקום מסוים "לאן ללכת". הסימן **:1** שנמצא לפני הפקודה **Randomize** משחק את תפקיד ה"שלט". כלומר, אם בחפיסה השמאלית אין קלף עם מספר זה (אפשרות **Else**), מתבצעת פקודת העברה ללא תנאי אל סימן **:1**, לאחר מכן הקלפים מתערבבים שוב, מספר מקרי נבחר שוב, ושוב מתבצעת בדיקה האם קלף עם מספר זה נמצא בחפיסה השמאלית וכו'...

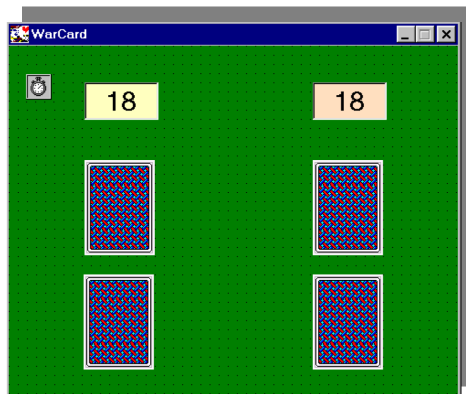
נמשיך את עיצוב הטופס

לאחר חלוקת הקלפים חצי-חצי בין השחקן למחשב עלינו לכסות אותם ב"מכסים".

1. ניצור על הטופס שתי מסגרות **Image** עם שמות **img_DeckComp** ו-**img_DeckPlayer** ונכניס אליהן את הקובץ **deck.bmp** מהתיקיה ..Images\Card Game\Others שבתקליטור המצורף (מאפיין **Picture**).
2. נשים את "המכסה" **img_DeckComp** על החפיסה השמאלית (**Top = 3000**, **Left = 1000**), ואת "המכסה" **img_DeckPlayer** במקום שצריכה להיות החפיסה הימנית (**Top = 3000**, **Left = 4000**).
3. נצייר שני תוויות **Labels** עם שמות **lbl_ScoreComp** ו-**lbl_ScorePlayer** ונמקם אותן מעל החפיסה הימנית והחפיסה השמאלית, כפי שנראה בתמונה.
4. לשתי התוויות נקבע **Caption - 18**, נגדיר את המאפיין **Alignment - Center** נקבע את גודל הגופן, את צבעו (מאפיינים **Font** ו-**ForeColor**) וגם את צבע התווית (מאפיין **BackColor**) ואת צורתה (מאפיין **BorderStyle**).
5. ניצור שתי מסגרות Images נוספות עם שמות **img_Deck1** ו-**img_Deck2** ונמקם אותן מעל שתי החפיסות. נקבע להן את הקואורדינטות ואת המאפיינים הבאים:

img_Deck1:	**Top = 1500, Left = 1000, Visible - False**
img_Deck2:	**Top = 1500, Left = 4000, Visible - False**

6. נציב טיימר (**Timer1**).



נתחיל משחק

כעת כאשר ממשק המשחק מוכן והקלפים מחולקים, אפשר להתחיל לשחק. השחקן והמחשב צריכים להוציא קלף אחד מהחפיסות שלהם. למטרה זו לוחץ השחקן בעזרת העכבר על "המכסה" הימני - **img_DeckPlayer**. הקלפים שהוצאו צריכים להימצא מעל החפיסות בגובה **Top = 1500**.

1. נכתוב את הפרוצדורה ללחיצה בעכבר על המסגרת-"מכסה" **img_DeckPlayer**:

```
Private Sub img_DeckPlayer_Click()
```

2. מכל חפיסה צריך להוציא קלף אחד ואת שניהם לשלוח למעלה:

```
img_Card(i).Top = 1500
img_Card(j).Top = 1500
```

נבדוק כיצד עובדת הפרוצדורה. אם נפעיל את התוכנית מספר פעמים ברציפות, נראה שמהחפיסה הימנית ייצאו קלפים לעיתים קרובות יותר, אך לפעמים הם ייצאו משתי החפיסות, מדוע?

ננסה להסביר את התופעה. עוד לפני שלחצנו על "המכסה", בפרוצדורת טעינת הטופס **Private Sub Form_Load()** התחלקו הקלפים במספרים באופן אקראי (**Randomize**). חצי מהם נשאר בחפיסה השמאלית וחצי עבר לחפיסה הימנית. הערך האחרון של מספר-האינדקס המקרי **i** תואם תמיד לקלף בחפיסה ***הימנית***. לכן, הפקודה **img_Card(i).Top = 1500** מתבצעת תמיד, ומהחפיסה הימנית יוצא קלף עם אינדקס השווה למספר המקרי **i**.

אך, למעשה, יוצאים שני קלפים מהחפיסה הימנית ולא אחד! אחד מהם "עולה" על השני ולכן נדמה לנו שיצא רק קלף אחד. רוצים לבדוק? נשנה את **img_Card(j).Top** ל-**1000** ונשתכנע שהדבר הוא כך:

```
Private Sub img_DeckPlayer_Click()
        img_Card(i).Top = 1500
        img_Card(j).Top = 1000
End Sub
```

אז מדוע לפעמים יוצאים הקלפים משתי החפיסות?

הכל תלוי במשתנה-מונה **j**, ששווה ל-**0**. איפה נמצא הקלף עם משתנה-מונה אפסי (**img_Card(0)**: בחפיסה הימנית או בחפיסה השמאלית? אם הקלף נשאר בחפיסה השמאלית, הקלפים יוצאים משתי החפיסות; אם הוא שייך לשחקן, או במילים אחרות עבר לחפיסה הימנית, שני הקלפים ייצאו מחפיסה זו. כיצד נפתור את הבעיה?

1. קודם כל צריך לסדר שמשתנה **j** יקבל עם כל לחיצה חדשה, ערך אקראי כלשהו:

```
Randomize
    j = (Rnd * 35)
```

2. כעת יש להגביל את חיפוש הקלפים עם אינדקס **i** (קלפי המחשב) בחפיסה השמאלית בלבד:

```
If img_Card(i).Left = 1000 Then
      img_Card(i).Top = 1500
```

3. אם אין בחפיסה השמאלית קלף עם מספר זה (בתחילת המשחק קלף **i** תמיד יהיה בחפיסה הימנית), יש להוסיף אחד למספר זה, ולהתחיל שוב בבדיקה:

```
Else
    i = i + 1
        GoTo 1
```

4. לסימן **1:** שאליו מתבצע המעבר, נשלב תנאי נוסף:

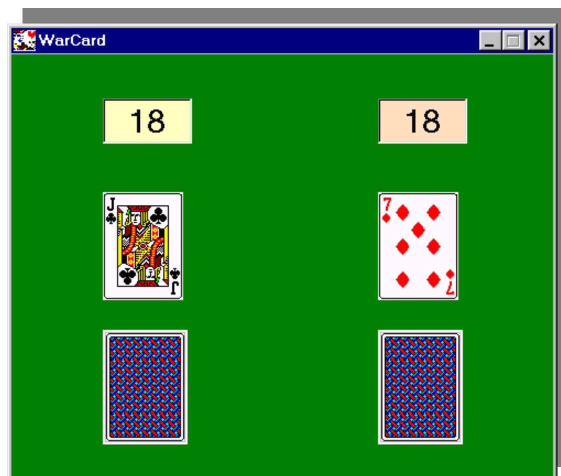
```
1:   If i = 36 Then i = 0
```

כך נוכל למנוע את הופעת המספר (אינדקס) ששווה ל-**36**. הרי ייתכן שברגע כלשהו של המשחק נקבל ערך **i = 35**. אם בחפיסה השמאלית לא יהיה קלף עם מספר זה, למספר **35** יתווסף **1** ונקבל מרכיב מערך שלא קיים - קלף שערך האינדקס שלו הוא **36**.

5. את הפקודות האלו נכתוב גם לקלפים שבחפיסה הימנים שלהם מונה **j**. כתוצאה נקבל טקסט מלא, שבו צריך להוסיף את פקודת הדלקת הטיימר:

```
Private Sub img_DeckPlayer_Click()
1:   If i = 36 Then i = 0
     If img_Card(i).Left = 1000 Then
        img_Card(i).Top = 1500
     Else
       i = i + 1
       GoTo 1
     End If
     Randomize
     j = (Rnd * 35)
2:   If j = 36 Then j = 0
     If img_Card(j).Left = 4000 Then
       img_Card(j).Top = 1500
     Else
       j = j + 1
       GoTo 2
     End If
     Timer1.Enabled = True
End Sub
```

נבדוק את עבודת הפרוצדורה. אם נלחץ **פעם אחת** על "המכסה" הימני, מעל כל אחת משתי החפיסות יופיע קלף אחד. כדי לבדוק את התוכנית שנית, יש לסגור את הטופס ולהפעיל אותה שוב. אם הפרוצדורה נרשמה נכון, בכל הפעלה חדשה צריכים להופיע קלפים שונים.



למי שייך הקלף המנצח?

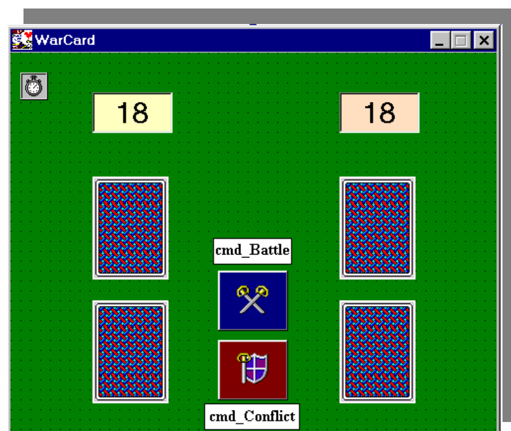
עכשיו על הטיימר לבדוק למי שייך הקלף הגדול יותר. קודם כל ניזכר שלכל קלף הוענק תג. כלומר, הטיימר צריך להשוות את התגים. הטיימר יודע לבצע את הדבר הזה מייד, אך רצוי ליצור רושם כאילו המחשב חושב שנייה, לכן במאפייני הטיימר נגדיר **Interval - 1000** ונכבה את הטיימר (מאפיין **Enabled - False**).

במשחק שלנו עלולים להופיע שני מצבים:

- הקלפים אינם שווים, ואז קלף אחד "אוכל" את השני.
- קלפים שווים ואז פורצת "מלחמה".

במקרה "מלחמה" נצטרך לשני לחצנים עם שמות **cmd_Conflict** ו-**cmd_Battle**. תמונות ללחצנים אלה תמצא בתיקיה: ..Images\Card Game\Others.

נמקם את הלחצנים בין שתי החפיסות ונעניק לשניהם את המאפיין **Visible - False**.



1. לחלק **General** נוסיף משתנה - מונה הנקודות **Score**:

```
Dim Score As Integer
```

2. נפתח פרוצדורה בשביל הטיימר:

```
Private Sub Timer1_Timer()
```

3. נגדיר את התנאי: אם הקלפים אינם שווים זה לזה.

```
If img_Card(i).Tag <> img_Card(j).Tag Then
```

4. במקרה זה נציב במונה הנקודות 1:

```
Score = 1
```

5. בתוך התנאי נגדיר תנאי שני - כאשר קלף המחשב "מושך" את קלף השחקן:

```
If       img_Card(i).Tag > img_Card(j).Tag Then
```

6. במקרה זה מתבצעת סדרת הפקודות:

```
        ' קלף השחקן עובר לחפיסה השמאלית
        img_Card(j).Left = 1000
        ' קלף המחשב וקלף השחקן - שניהם יורדים למטה
        img_Card(i).Top = 3000
        img_Card(j).Top = 3000
        ' מספר הקלפים של השחקן פוחת ב-1 (Score = 1)
        Player_cards = Player_cards - Score
        ' בתווית מעל קלפי השחקן יופיע מספר הקלפים שלו
        lbl_Player.Caption = Player_cards
        ' בתווית מעל קלפי המחשב יופיע מספר קלפי המחשב
        lbl_Computer.Caption = 36 - Player_cards
```

7. קיימת גם אלטרנטיבה - **Else**, כאשר קלף השחקן "מכה" את קלף המחשב. במקרה זה קלף המחשב עובר ימינה, והקלף מצטבר לשחקן:

```
        Else
            img_Card(i).Left = 4000
            img_Card(i).Top = 3000
            img_Card(j).Top = 3000
            Player_cards = Player_cards + Score
            lbl_Player.Caption = Player_cards
            lbl_Computer.Caption = 36 - Player_cards
        End If
```

8. בשלב זה נוסיף אלטרנטיבה - **Else** לתנאי הראשון. המשמעות היא שקלפי המחשב והשחקן שווים. במקרה זה, נציג את הלחצן **cmd_Conflict** ונקבע ל"מכסה" את המאפיין **Enabled** כ-**False**, משום שאסור ללחוץ עליו יותר.

```
 Else
   cmd_Conflict.Visible = True
   img_DeckPlayer.Enabled = False
 End If
```

9. בסוף הפרוצדורה נגדיל את המשתנים **i** ו-**j** באחד, כדי שעם לחיצה הבאה על "המכסה" הקלפים יתחלקו אחרת. בנוסף, נדליק את הטיימר.

כך נראה הטקסט המלא של הפרוצדורה שלנו:

```
Private Sub Timer1_Timer()
   If img_Card(i).Tag <> img_Card(j).Tag Then
      Score = 1
      If img_Card(i).Tag > img_Card(j).Tag Then
        img_Card(j).Left = 1000
        img_Card(i).Top = 3000
        img_Card(j).Top = 3000
        Player_cards = Player_cards - Score
        lbl_Player.Caption = Player_cards
        lbl_Computer.Caption = 36 - Player_cards
      Else
        img_Card(i).Left = 4000
        img_Card(i).Top = 3000
        img_Card(j).Top = 3000
        Player_cards = Player_cards + Score
        lbl_Player.Caption = Player_cards
        lbl_Computer.Caption = 36 - Player_cards
      End If
  Else
      cmd_Conflict.Visible = True
      img_DeckPlayer.Enabled = False
```

```
    End If
    i = i + 1
    j = j + 1
    If i = 36 Then i = 0
    If j = 36 Then j = 0
    Timer1.Enabled = False
End Sub
```

נשלח את הקלפים למלחמה

1. נוסיף לרשימת המשתנים מונה **k**, שיסייע לספור את הקלפים הנשלחים ל"מלחמה":

```
Dim k As Integer
```

2. נכתוב לולאה בפרוצדורה **Private Sub cmd_Conflict_Click()**:

```
Private Sub cmd_Conflict_Click()
For k = 1 To 3
    Score = Score + 1
1:      If i = 36 Then i = 0
        If img_Card(i).Left = 1000 Then
            img_Card(i).Top = 1500
            img_Deck1.Visible = True
        Else
            i = i + 1
            GoTo 2
        End If
2:      If j = 36 Then j = 0
        If img_Card(j).Left = 4000 Then
           img_Card(j).Top = 1500
           img_Deck2.Visible = True
        Else
            j = j + 1
            GoTo 2
        End If
        i = i + 1
        j = j + 1
        If i = 36 Then i = 0
        If j = 36 Then j = 0
    Next
        If img_Card(i).Tag <> img_Card(j).Tag Then
       cmd_Conflict.Visible = False
       cmd_Battle.Visible = True
    End If
End Sub
```

ההיגיון של הפרוצדורה מזכיר לנו את הפרוצדורה ל"מכסה" **img_DeckPlayer**. ההבדל הוא בזה שכאן מספר הקלפים הנשלחים ל"מלחמה" תלוי במונה **k**. חוץ מזה, הקלפים הנשלחים למעלה מכוסים על ידי "מכסים" (**img_Deck1** ו-**img_ Deck2**).

אם הקלפים אינם שווים זה לזה, הלחצן **Conflict** נעלם ומופיע הלחצן **Battle**. אם הקלפים האחרונים שוב שווים, על השחקן ללחוץ שוב על הלחצן **Conflict**, והפרוצדורה תחזור שוב על עצמה.

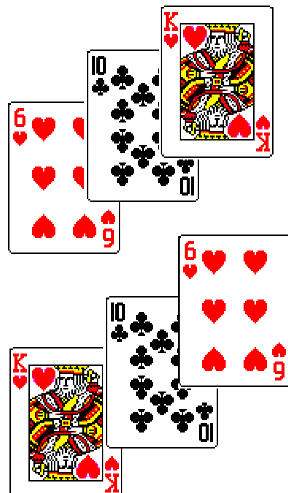
נשים לב למשתנה **Score**. בכל חזרה (איטרציה) יתווסף אחד (**Score = Score + 1**). כך שבסיום עבודת הפרוצדורה, המשתנה **Score** יגדל במספר השווה למספר הקלפים שנשלחו ל"מלחמה". מי ירוויח את הנקודות הנוספות האלה, השחקן או המחשב, תלוי בעבודת הלחצן **Battle**.

הקלפים יוצאים למלחמה

ובכן הקלפים מוכנים למלחמה. השאלה מי ינצח, קלף **i** או קלף **j** תלויה בערך התג (**Tag**). אם נלחץ על הלחצן Battle, מכסי הקלפים ייפתחו ונראה את תוצאת הקרב:

```
Private Sub cmd_Battle_Click()
img_Deck1.Visible = False
img_Deck2.Visible = False
```

אך כאן מסתתר "באג" קטן, אפילו לא "באג" של ממש, אלא תוצאת העיצוב הלא מוצלח שלנו. העניין הוא שתוצאת הקרב תלויה בתגי הקלפים **i** ו-**j** שנשלחו למעלה **אחרונים**, אך יחד עם זה, מיקום הקלפים על הטופס תלוי בסדר שבו הכנסנו את התמונות למסגרות **img_Card**. לכן, הקלף האחרון שנשלח למעלה לא חייב לבוא מעל שאר הקלפים, הוא יכול להימצא גם מתחתם, דבר זה תלוי באינדקס **img_Card**. כתוצאה מכך, מעבודת הפרוצדורה **Private Sub cmd_Battle_Click()** אפשר להתרשם לפעמים שהקלף הקטן מושך את הקלף הגדול יותר.



למשל, אם התמונות הוכנסו למסגרות בסדר: "שש", "עשר", "מלך", המיקום שלהם בטופס יהיה כמו שנראה בתמונה, כלומר: "שש" יימצא למטה, מעליו "עשר" ולמעלה "מלך". הקלף שאינדקס שלו גדול יותר יימצא מעל הקלף עם האינדקס הקטן יותר.

עכשיו נתאר לעצמנו, שאותם הקלפים נשלחו ל"מלחמה", אך סדר הצגתם היה תלוי בערך המשתנה המקרי **i** או **j**. אפשרי מצב שהקלף הראשון שמוצג הוא "מלך", אחריו "עשר" ובסוף "שש". לכן, לפי כללי משחק אמיתי, אנו צריכים למקם את הקלפים כפי שנראה בתמונה השנייה.

בכל זאת הדבר אינו כך. הקלפים שנשלחו ל"מלחמה" יסתדרו לפי סדר האינדקסים שלהם ללא קשר לסדר הצגתם.

לא כדאי לשים לב לכך, אחרת נצטרך לבנות את הפרויקט שלנו מחדש, ולכתוב פרוצדורות הרבה יותר מסובכות מאלו שכתבנו עד עכשיו.

הפעולה הבאה שמתבצעת על ידי לחצן **Battle**, היא השוואת תגי הקלפים הנשלחים ל"מלחמה" אחרונים:

```
If img_Card(i).Tag > img_Card(j).Tag Then
```

1. אם תג (**Tag**) הקלף של המחשב גדול יותר, המחשב צריך למשוך מהשחקן את כל הקלפים שנשלחו ל"מלחמה". איך נעשה זאת? הרי התוכנית זוכרת רק את מספרי הקלפים האחרונים **i** ו-**j**, שאר הקלפים נשכחו. לא קשה לכתוב את הפרוצדורה להעברת קלף **j** לחפיסה **i** - פרוצדורה כזו קיימת כבר בתוכניתנו, בעבודת הטיימר, אבל כיצד נעביר לחפיסה זו את הקלפים שנשארו למעלה?

   קודם כל, נבצע חיפוש של הקלפים האלה בין כל הקלפים שבמשחק:

```
For k = 0 To 35
```

2. אחר נגדיר את הקלף שעליו ידוע בוודאות שהוא נמצא בחלק הימני של הטופס **img_Card(k).Left = 4000** וגם נשלח למעלה **img_Card(k).Top = 1500**:

```
  If img_Card(k).Left = 4000 And _
    img_Card(k).Top = 1500 Then
```

3. כאשר קלף זה נמצא יש להזיז אותו שמאלה, אך להגדיר את הזזתו בקואורדינטת החפיסה השמאלית **img_Card(k).Left = 1000**. פרוצדורת הזזת הקלף נרשמת בלולאה:

```
    Do Until img_Card(k).Left = 1000
        img_Card(k).Move img_Card(k).Left - 1
     Loop
```

4. לאחר כשהקלף עובר לחלק השמאלי של הטופס, נוריד אותו אל חפיסת המחשב:

```
  img_Card(k).Top = 3000
```

5. נמצא את הקלף הראשון מקלפי המחשב, שנשלחו למעלה. איננו זקוקים להם יותר, לכן נוריד גם אותם למטה לחפיסת המחשב:

```
    If img_Card(k).Left = 1000 And _
          img_Card(k).Top = 1500 Then
        img_Card(k).Top = 3000
     End If
```

6. בזה הסתיימה עבודת החזרה (איטרציה) הראשונה של הלולאה, וניתן לעבור לחזרה הבאה:

```
Next
```

7. עלינו להגדיל את חשבון הנקודות לטובת המחשב ולהראות אותו בתוויות:

```
Player_cards = Player_cards - Score
lbl_Player.Caption = Player_cards
lbl_Computer.Caption = 36 - Player_cards
```

8. ואם יצא הפוך וקלף השחקן משך את קלף המחשב. אז לתנאי **If img_Card(i).Tag > img_Card(j).Tag Then** יש אלטרנטיבה:

```
Else
```

במקרה זה יוצא הכל להיפך: קלפי המחשב עוברים לחפיסת השחקן, וחשבון הקלפים משתנה לטובתו. בסוף הפרוצדורה נסגר תנאי **If**, הלחצן **Battle** מוסתר והפעילות "המכסה" **img_DeckPlayer** מחודשת.

כך נראה הפרוצדורה בשלמותה:

```
Private Sub cmd_Battle_Click()
img_Deck1.Visible = False
img_Deck2.Visible = False
If img_Card(i).Tag > img_Card(j).Tag Then
  For k = 0 To 35
    If img_Card(k).Left = 4000 And _
       img_Card(k).Top = 1500 Then
         Do Until img_Card(k).Left = 1000
           img_Card(k).Move img_Card(k).Left - 1
         Loop
         img_Card(k).Top = 3000
    End If
    If img_Card(k).Left = 1000 And_
       img_Card(k).Top = 1500 Then
        img_Card(k).Top = 3000
     End If
  Next
  Player_cards = Player_cards - Score
  lbl_Player.Caption = Player_cards
  lbl_Computer.Caption = 36 - Player_cards
Else
  For k = 0 To 35
    If img_Card(k).Left = 1000 And _
        img_Card(k).Top = 1500 Then
         Do Until img_Card(k).Left = 4000
```

```
            img_Card(k).Move img_Card(k).Left + 1
          Loop
          img_Card(k).Top = 3000
      End If
      If img_Card(k).Left = 4000 And _
       img_Card(k).Top = 1500 Then
          img_Card(k).Top = 3000
      End If
   Next
   Player_cards = Player_cards + Score
   lbl_Player.Caption = Player_cards
   lbl_Computer.Caption = 36 - Player_cards
End If
cmd_Battle.Visible = False
img_DeckPlayer.Enabled = True
End Sub
```

סיום המשחק

המשחק שלנו הוא הפשוט מכל משחקי הקלפים, לכן הוא מסתיים בניצחון מלא של אחד הצדדים. זאת אומרת, כשלשחקן לא נשאר אף קלף, או להיפך, הוא לקח מהמחשב את כל **36** הקלפים. ניתן לבטא תנאי זה בפרוצדורה הבאה:

```
If Player_cards <= 0 Then Form1.Caption = "You loose!"
If Player_cards >= 36 Then Form1.Caption = "You won!"
```

איפה נרשום את הפרוצדורה הזו, מי ממרכיבי תוכניתנו יכול לפקח האם מתבצע אחד מהתנאים הללו?

נשים לב שבתוויות **lbl_Player** ו-**lbl_Computer** מתרחש **שינוי** קבוע (**Change**) של ערך המשתנה **Player_cards**. נקשור את הפרוצדורה שלנו לאירוע זה. למטרה זו נציב קודם את הפרוצדורה **Private Sub lbl_Player_Click()** ואחר, בחלק הימני העליון של חלון הקוד, נבחר את האירוע **Change**:

```
 Private Sub lbl_Player_Change()
If Player_cards >= 36 Then Form1.Caption = "You won!"
If Player_cards <= 0 Then Form1.Caption = "You loose!"
End Sub
```

עכשיו התוכנית שלנו שלמה ואפשר לשחק ב"מלחמה". כדי שהמשחק יהיה מעניין ומושך יותר, אפשר לשנות את עיצוב הטופס, למשל, כפי שעשו תלמידי בית-הספר שלנו NetArt. כבסיס לעיצוב הם בחרו במשחק המחשב הפופולרי בזמנו - Warcraft-2.

תמונת פתיחת המשחק. כאן אנו משתמשים בשני לחצני אפשרויות - **Option Button**. כתלות בבחירת הלחצן, "המסכה" **img_DeckPlayer** יכול להימצא בחלק השמאלי או בחלק הימני (**img_DeckPlayer.Left = 1000** או **img_DeckPlayer.Left = 1000**).

ממשק המשחק. בעיצוב שימשו תמונות המשחק Warcraft-2 שנערכו בעזרת התוכנית Map Editor.

טקסט התוכנית המלא

```
Dim i, j, k As Integer
Dim War_Cards, Player_cards, Score As Integer

Private Sub Form_Load()
  Do Until Player_Cards = 18
1:     Randomize
       i = (Rnd * 35)
       If img_Card(i).Left = 1000 Then
         img_Card(i).Left = 4000
         Player_Cards = Player_Cards + 1
       Else
         GoTo 1
       End If
  Loop
 End Sub

Private Sub img_DeckPlayer_Click()
1:  If i = 36 Then i = 0
    If img_Card(i).Left = 1000 Then
       img_Card(i).Top = 1500
    Else
      i = i + 1
      GoTo 1
    End If
    Randomize
    j = (Rnd * 35)
2:  If j = 36 Then j = 0
    If img_Card(j).Left = 4000 Then
      img_Card(j).Top = 1500
    Else
     j = j + 1
     GoTo 2
    End If
    Timer1.Enabled = True
End Sub
```

```
Private Sub Timer1_Timer()
   If img_Card(i).Tag <> img_Card(j).Tag Then
      Score = 1
      If img_Card(i).Tag > img_Card(j).Tag Then
        img_Card(j).Left = 1000
        img_Card(i).Top = 3000
        img_Card(j).Top = 3000
        Player_cards = Player_cards - Score
        lbl_Player.Caption = Player_cards
        lbl_Computer.Caption = 36 - Player_cards
      Else
        img_Card(i).Left = 4000
        img_Card(i).Top = 3000
        img_Card(j).Top = 3000
        Player_cards = Player_cards + Score
        lbl_Player.Caption = Player_cards
        lbl_Computer.Caption = 36 - Player_cards
      End If
   Else
      cmd_Conflict.Visible = True
      img_DeckPlayer.Enabled = False
   End If
   i = i + 1
   j = j + 1
   If i = 36 Then i = 0
   If j = 36 Then j = 0
   Timer1.Enabled = False
End Sub
```

```
Private Sub cmd_Conflict_Click()
For k = 1 To 3
   Score = Score + 1
1:    If i = 36 Then i = 0
      If img_Card(i).Left = 1000 Then
         img_Card(i).Top = 1500
         img_Deck1.Visible = True
      Else
         i = i + 1
         GoTo 2
      End If
2:    If j = 36 Then j = 0
      If img_Card(j).Left = 4000 Then
        img_Card(j).Top = 1500
```

```
            img_Deck2.Visible = True
        Else
            j = j + 1
            GoTo 2
        End If
        i = i + 1
        j = j + 1
        If i = 36 Then i = 0
        If j = 36 Then j = 0
    Next
        If img_Card(i).Tag <> img_Card(j).Tag Then
        cmd_Conflict.Visible = False
        cmd_Battle.Visible = True
    End If
End Sub
```

```
Private Sub cmd_Battle_Click()
img_Deck1.Visible = False
img_Deck2.Visible = False
If img_Card(i).Tag > img_Card(j).Tag Then
  For k = 0 To 35
    If img_Card(k).Left = 4000 And _
       img_Card(k).Top = 1500 Then
          Do Until img_Card(k).Left = 1000
            img_Card(k).Move img_Card(k).Left - 1
          Loop
          img_Card(k).Top = 3000
     End If
     If img_Card(k).Left = 1000 And_
        img_Card(k).Top = 1500 Then
         img_Card(k).Top = 3000
      End If
  Next
  Player_cards = Player_cards - Score
  lbl_Player.Caption = Player_cards
  lbl_Computer.Caption = 36 - Player_cards
Else
  For k = 0 To 35
     If img_Card(k).Left = 1000 And _
         img_Card(k).Top = 1500 Then
          Do Until img_Card(k).Left = 4000
            img_Card(k).Move img_Card(k).Left + 1
          Loop
```

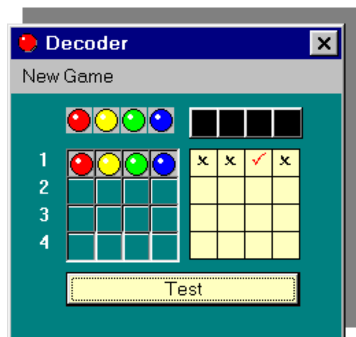
```
        img_Card(k).Top = 3000
    End If
    If img_Card(k).Left = 4000 And _
     img_Card(k).Top = 1500 Then
        img_Card(k).Top = 3000
     End If
  Next
  Player_cards = Player_cards + Score
  lbl_Player.Caption = Player_cards
  lbl_Computer.Caption = 36 - Player_cards
End If
cmd_Battle.Visible = False
img_DeckPlayer.Enabled = True
End Sub
```

```
Private Sub lbl_Player_Change()
        If Player_cards >= 36 Then Form1.Caption = "You won!"
        If Player_cards <= 0 Then Form1.Caption = "You loose!"
End Sub
```

6 משחק "קוד סודי"

כללי המשחק

משחק זה דומה באופן כללי ל"בול פגיעה". בלוח המשחק נמצאים ארבעה כדורים צבעוניים. מתחתם טבלה עם ארבע משבצות בכל שורה. בצד הימני העליון של הממשק יש ארבעה ריבועים שחורים. בהם רשום "הקוד הסודי" - כדורים צבעוניים אקראיים. הסדר וצבע הכדורים בריבועים השחורים אינם ידועים (אותו צבע יכול להופיע מספר פעמים). יש לגרור בעזרת העכבר את הכדורים למשבצות כדי למלא את השורות.



אם סדר הכדורים תואם לקוד הסודי - המשחק נגמר, אך הסתברות של ניחוש כזה קטנה מדי. בדרך כלל אפשר רק לנחש מיקום של אחד או מספר כדורים. כדי לבדוק את תוצאת הניחוש, יש ללחוץ על הלחצן "**Test**". בצד ימין של הטבלה יופיעו סימנים: ניחוש נכון - סימן ✓ אדום וניחוש לא נכון - סימן × שחור.





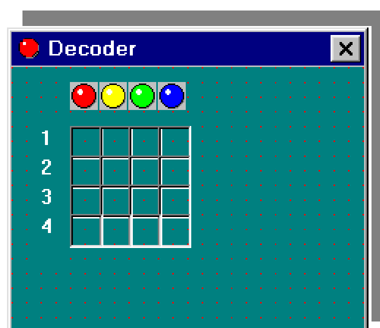
יתר כללי המשחק מובנים לחלוטין: יש לאתר את מיקום וצבע הכדורים שלא נוחשו (תקבלו תשובה × שחור). כאן הכל תלוי בכושר התפישה של השחקן משום שיש רק 4 נסיונות ניחוש. אם השורות מולאו והקוד לא נוחש, תופיע הודעה שכל הנסיונות תמו. אם הקוד נוחש, הריבועים השחורים ייפתחו ויציגו את הכדורים שהוסתרו.

הכנה לעיצוב הפרויקט

1. ניצור תיקיה חדשה בשם **Decoder**.
2. נעביר מהתיקיה Images, שנמצאת בתקליטור המצורף לספר ארבע תמונות: **red.bmp**, **yellow.bmp**, **green.bmp** ו-**blue.bmp**.
3. בדיוק כפי שעשינו בפרויקט "בינגו", נקרא לקבצים אלה בשמות חדשים: קובץ **red.bmp** ייקרא **0**, קובץ **yellow.bmp - 1**, **green.bmp - 2** ו-**blue.bmp - 3**. לקבצים **0**, **1**, **2** ו-**3** אין צורך בסיומת!

ממשק המשחק

1. גודל הטופס המועדף: **Width - 3400** ו-**Heigth - 3000**.
2. נסיר משורת כותרת הטופס לחצנים מיותרים (**BorderStyle - 1-Fixed Single**) נקליד את שם הטופס (**Caption - Decoder**) ונוסיף תמונת כדור מהתיקיה Images שבתקליטור המצורף (מאפיין **Icon - Decoder.ico**).
3. נמקם על הטופס ארבע מסגרות - **Image** בשמות **img_Red**, **img_Yellow**, **img_Green** ו-**img_Blue**.
4. בכל מסגרת נשתול את התמונה המתאימה, ולכולן נקבע גודל **285** X **285**:
   - **img_Red** - קובץ **0** (הכדור האדום);
   - **img_Yellow** - קובץ **1** (הכדור הצהוב);
   - **img_Green** - קובץ **2** (הכדור הירוק);
   - **img_Blue** - קובץ **3** (הכדור הכחול).
5. בכל מסגרת נשנה בחלון המאפיינים את **Tag** למספר התואם למספר הקובץ:
   - **img_Red - Tag 0;**
   - **img_Yellow - Tag 1;**
   - **img_Green - Tag 2;**
   - **img_Blue - Tag 3.**

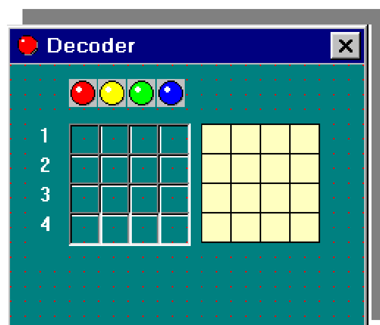


6. ניצור מערך מ-16 מסגרות **Image** (כל מסגרת בגודל 285 X 285) בשמות **img_Dec(0) ... img_Dec(15)** ונמקם אותן מתחת לתמונות כטבלה משבצות **4 × 4**. כדי שהיקף המסגרות יודגש, נציב את המאפיין **BorderStyle** במצב **1-Fixed Single**. את המאפיין **DragMode** נציב במצב **1-Automatic**.

7. יש למספר את השורות באמצעות תוויות רגילות - **Labels**, שבהן נכתוב מספרים (**1**, **2**, **3** ו-**4**).

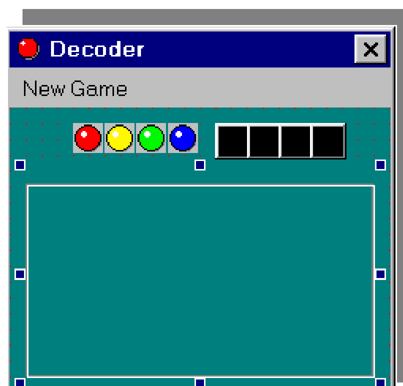
8. ליד טבלת המסגרות **img_Dec** ניצור אותה טבלה - מערך **4 × 4** מתוויות (**Labels**) בשמות **lbl_Order(0)...lbl_Order(15)**, כל אחת בגודל 285 X 285).

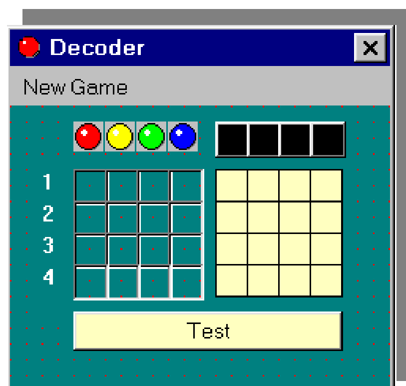
9. את המאפיין **BorderStyle** נקבע במצב **1-Fixed Single**, אך כדי שהתוויות תיראנה שטוחות נקבע את המאפיין **Appearance** במצב **Flat - 0**. אפשר לצבוע אותן בצבע עמום כלשהו (מאפיין **BackColor**). נקבע מאפיין **Stretch** כ-**False**.



10. נסמן בעכבר את כל המשבצות **lbl_Order** ונקבע את מאפיין **Alignment - 2-Center**. במאפיין **Font** נבחר גופן Wingdings 2. אנו זקוקים לו, כדי שבמקום אותיות רגילות יופיעו הסימנים ✗ ו-✓.

11. ניצור מערך מלחצנים רגילים (**CommandButton**) עם המידות **Width - 285** ו-**Heigth - 285**. ללחצנים נקרא בשמות **cmd_Code(0)...cmd_Code(3)**, בחלון המאפיינים נקבע **Style - Graphical** ונצבע את הלחצנים בצבע כהה. ללחצנים נקבע **Enabled - False** ונמקם את מערך הלחצנים מעל הטבלה **lbl_Order** ומימינה.





12. בחלק התחתון נוסיף לחצן **Test** בשם **cmd_Test**.

13. בעורך התפריטים ניצור תפריט **New Game** בשם **mnu_New**.

14. נוסיף לטופס מסגרת - **Frame**, שתכסה את שני מערכי המשבצות ואת לחצן **Test**. המסגרת אינה אמורה להיראות בתחילת המשחק, כלומר, נקבע את מאפיין **Visible** במצב **False**.

15. במרכז מסגרת הפריים נצייר תווית - **Label** בשם **lbl_Message** ונקבע בשבילה את המאפיין **Alignment - 2-Center**.

הממשק שלנו מוכן. אפשר לגשת לכתיבת המשחק.

ניצור את הקוד הסודי

1. ראשית יש להגדיר בחלק **General** שני משתנים - מונים:

```
Dim i, j  As Integer
```

2. לשם יצירת הקוד נשתמש בשיטה המוכרת לנו מכבר: בעת טעינת הטופס (אירוע **Form Load**) נוסיף למערך הלחצנים **cmd_Code** את מספרי הקבצים - תמונות באופן אקראי (**Randomize**), כפי שעשינו בפרויקט משחק "בינגו", כמו כן נשנה את מאפיין **Tag** של הלחצנים למספר התואם את מספר קובץ התמונה.

```
Private Sub Form_Load()
   For j = 0 To 3
      Randomize
      i = Int(4 * Rnd)
      cmd_Code(j).DownPicture = LoadPicture(i)
      cmd_Code(j).Tag = i
   Next
  End Sub
```

בפרוצדורה זו השתמשנו במאפיין **DownPicture** במקום המאפיין הרגיל **Picture**, כדי להוסיף את התמונה ללחצן. את התמונה שהוכנסה כ-**DownPicture** ניתן לראות רק כאשר נלחץ על הלחצן. אך משום שקבענו ללחצנים את המאפיין **Enabled - False**, לא נוכל ללחוץ עליהם ולראות את התמונות. הקוד המורכב מכדורים צבעוניים ממוספרים מ-**0** עד **3**, באמת הפך סודי!

כיצד להציג את התמונות?

אנו רגילים שב-Windows ניתן לתפוס כל מרכיב במסך באמצעות העכבר, "לגרור" ו"לשחרר" במקום אחר. טכניקה זו נקראת **Drag and Drop** - "גרור ושחרר". אנו נשתמש בטכניקה זו, כדי להעביר את הכדורים למשבצות הטבלה. ל- Visual Basic יש אירוע מיוחד למטרה זו שנקרא **DragDrop**.

בחלק השמאלי העליון של חלון הקוד נאתר את המרכיב **img_Dec**, בחלק הימני העליון נבחר את אירוע **DragDrop** ונכתוב את הפרוצדורה:

```
Private Sub img_Dec_DragDrop(Index As Integer,
      Source As Control, As Single, Y As Single)
            img_Dec(Index).Picture = Source.Picture
            img_Dec(Index).Tag = Source.Tag
            cmd_Test.Enabled = True
End Sub
```

בפרוצדורה זו Visual Basic מכניסה באופן אוטומטי את **Source** כמשתנה מסוג **Control**. מה זאת אומרת?

אנו כבר מכירים משתנים מסוג מספרים שלמים (**Integer**) ומשתנים מסוג לוגי (**Boolean**). אנו יודעים שאם למשל נגדיר משתנה בשם **a** כ-**Integer**, נוכל לקבוע למשתנה זה ערכים מתחום מספרים שלמים, למשל 1, 100, 345 וכו'. אם נגדיר את **a** כ-**Boolean**, הוא יוכל לקבל שני ערכים: "אמת" - **True** ו"שקר" - **False**.

אם נגדיר משתנה **a** כ-**Control** הוא יוכל לקבל ערכים של פקד (**Controls**), כמו לחצן פקודה (**CommandButton**), מסגרת-**Image** וכדומה. כמובן, לא מדובר בכך שהאות a הופכת לפתע ללחצן או לתמונה. מייד נראה כיצד הדבר מתרחש באמת.

למעשה, כאשר אנו מגדירים משתנה בתוכנית, אנו "מזמינים" בשבילו מקום בזיכרון המחשב. אנו מודיעים למחשב שהתוכנית תשתמש במשתנה בשם, בסוג ובתלות האלה, והמחשב "מסמן" בזיכרון תא בגודל מסוים. כאשר אנו משתמשים בשם המשתנה, הוא "זוכר" את המיקום (כתובת בזיכרון) של תא זה, כדי לקרוא ממנו כל פעם ערך מוחשי (קונקרטי) אחר של המשתנה.

אז מה משמעות העבודה עם משתנים? אם כתוצאה מעבודת התוכנית קיבלנו **a = 2**. ואנו צריכים להשתמש במספר 2, אנו מבקשים מהמחשב "להוציא" את המספר הזה מהתא בשם **a** - זה בעצם הכל!

עכשיו נתאר לעצמנו שבזיכרון המחשב "מוזמן" מקום - תא בשם **Source**. אם נלחץ בעכבר על פקד כלשהו, למשל על מסגרת **Image1**, אנו מכניסים לתא מידע על הפקד הנבחר. הביטוי **Source = Image1** אומר למחשב, שבתא בשם **Source** נשמר מידע על המסגרת **Image1**.

מייד נראה כיצד מתבצע עיקרון זה בפרוצדורה המסוימת שלנו.

כיצד עובדת הפקודה **img_Dec(Index).Picture = Source.Picture**?

אנו לוחצים בעכבר על אחד הכדורים-תמונות, למשל, **img_Red**. המשתנה **Source** מקבל את הערך השווה ל-**img_Red**, כלומר, שומר בתוך עצמו מידע על המסגרת-תמונה עם השם **img_Red**. לתמונה זו יש מאפיין **Picture** - דמות הכדור האדום שהוכנס למסגרת. המשתנה **Source** כאילו "יורש" מהמסגרת **img_Red** את המאפיין הזה, והכוונה של **Source.Picture** היא שהכדור האדום שלנו יבוא אחר המשתנה **Source** לכל מקום, בעיקר למשבצת בשם **img_Dec(Index)**.

המשבצת **img_Dec(Index)** היא אחת מ-16 משבצות המערך, שמקומה ומספרה בטבלה מוגדר על ידי האינדקס שלה (**Index**). אם אנו רוצים לגרור את הכדור האדום לשורה התחתונה, משבצת שנייה מימין, בבקשה! אינדקס המשבצת הזו שווה ל-14, ובהתאם לפקודה שלנו, משבצת זו תקבל את התמונה, שאנו נעביר לה ביחד עם המשתנה **Source**:

```
img_Dec(Index).Picture = Source.Picture
```

המשתנה **Source** גורר למשבצת הנבחרת את דמות הכדור הצבעוני וגם את מספר קובץ התמונה - **Tag**:

```
img_Dec(Index).Tag = Source.Tag
```

חוץ מזה, מתעורר לפעילות לחצן הבדיקה **Test**:

```
cmd_Test.Enabled = True
```

נפעיל את התוכנית. בעזרת העכבר ננסה "לגרור" את דמות הכדור למשבצת כלשהי ו"לשחרר" אותו. כך נראה את עבודת הפרוצדורה **DragDrop**.

כיצד לנחש קוד?

1. לחלק **General** נוסיף עוד שני משתנים:

```
Dim Click_Number, Cell_N As Integer
```

המשתנה **Click_Number** ישמש לחישוב מספר הלחיצות על הלחצן **Test**. לפי כללי המשחק יש רק ארבעה ניסיונות, כדי לנחש את הקוד הסודי, לכן המשחק ייעצר כאשר **Click_Number** יהיה שווה ל-**4**.

כדי לספור את משבצות הטבלה שלא באמצעות מספרי האינדקס שלהן, ניתן להשתמש במשתנה בשם **Cell_N**.

2. ננסה לנחש את הקוד בניסיון הראשון ונלחץ על הלחצן **Test**:

```
Private Sub cmd_Test_Click()
If      img_Dec(Cell_N).Tag = cmd_Code(0).Tag And _
        img_Dec(Cell_N + 1).Tag = cmd_Code(1).Tag And _
        img_Dec(Cell_N + 2).Tag = cmd_Code(2).Tag And _
        img_Dec(Cell_N + 3).Tag = cmd_Code(3).Tag Then
        For j = 0 To 3
          cmd_Code(j).Enabled = True
          cmd_Code(j).Picture =  cmd_Code(j).DownPicture
        Next
        img_Red.Visible = False
        img_Green.Visible = False
        img_Blue.Visible = False
        img_Yellow.Visible = False
        Frame1.Visible = True
        lbl_Message.Caption = "You have successfully solved it!"
Else
        If Click_Number = 4 Then
          Frame1.Visible = True
          lbl_Message.Caption = "You have no more attempts"
        End If
End If
```

נראה כיצד עובדת פרוצדורה זו:

אם **Tag** המשבצת הראשונה (**Cell_N = 0**) שווה ל-
Tag הלחצן הראשון (0) cmd_code **ו-**
Tag המשבצת השנייה (**Cell_N + 1**) שווה ל-
Tag הלחצן השני (1) cmd_code **ו-**
Tag המשבצת השלישית **(Cell_N + 2)**שווה ל-
Tag הלחצן השלישי (2) cmd_code **ו-**
Tag המשבצת הרביעית (**Cell_N + 3**) שווה ל-
Tag הלחצן הרביעי (3) cmd_code **אז**
בלולאה **של** j מ-0 **עד** 3
מפעילים את הלחצנים cmd_Code(j)
הופכים את התמונה לגלויה (מ-DownPicture מעבירים ל-Picture)
ממשיכים את הלולאה (ארבע חזרות)
מסתירים את התמונה **img_Red**
מסתירים את התמונה **img_Green**
מסתירים את התמונה **img_Blue**
מסתירים את התמונה **img_Yellow**
מכסים את חלק הטופס במסגרת **Frame1**
מציגים את הכתובת על התווית **lbl_Message**:
"You have successfully solved it!"
אחרת (במקרה, שהתנאי **אם** לא מתבצע)
אם מספר הלחיצות שווה ל-4 **אז**
מכסים את חלק הטופס במסגרת **Frame1**
מציגים את הכתובת על התווית **lbl_Message**:
"You have no more attempts"
מסיימים את בדיקת התנאי **אם**
מסיימים את בדיקת התנאי **אם** (הראשון)

קרוב לוודאי שהניסיון הראשון שלנו לא יצליח ולא ננחש את הקוד. צריך להשתמש בשלושת הניסיונות הבאים וגם במידע שנקבל במשבצות **lbl_Order** שבטבלה הימנית. נראה כיצד תעבוד לולאה עם המונה **i**:

```
For i = Cell_N To Cell_N + 3
If img_Dec(i).Tag = cmd_Code(i - Cell_N).Tag Then
   lbl_Order (i).ForeColor = vbRed
   lbl_Order (i).Caption = "P"
Else
   lbl_Order (i).Caption = "O"
End If
Next
Cell_N = Cell_N + 4
```

התנאי **For i = Cell_N To Cell_N + 3** מאפשר לנו לבדוק, האם ניחשנו נכון את הקוד בתוך שורת המשבצות. לפני הלחיצה הראשונה על הלחצן Test המשתנה **Cell_N** שווה ל-**0**, והבדיקה תתבצע על שורת המשבצות העליונה מ-**img_Dec(0)** עד **img_Dec(3)**.

לאחר סיום הבדיקה נוסיף את הספרה 4 למספר המשבצת: **Cell_N = Cell_N + 4**. הדבר מאפשר לארגן את בדיקת שורת המשבצות השנייה ובה האינדקסים מ-**4** עד **7**, לאחר מכן את השורה השלישית שבה האינדקסים מ-**8** עד **11** ולבסוף - בדיקת השורה האחרונה שבה האינדקסים מ-**12** עד **15**.

מה משמעות הבדיקה?

אם תג המשבצת עם אינדקס **i** שווה לתג הלחצן התואם **cmd_Code (i - Cell_N)**, יירשם במשבצת סימן **P**, התואם לסימן ✓ של הגופן Wingdings 2. צבע הסימן יהיה אדום - **lbl_Order (i).ForeColor = vbRed**.

אפשר להשתכנע שהפרש **i - Cell_N** מאפשר תמיד לקבל את אינדקסי המשבצות **cmd_Code** מ-**0** עד **3**:

כאשר 0 = **Cell_N**, אז **i - Cell_ N** משתנה מ-**0** עד **3**;

כאשר 4 = **Cell_N**, אז i משתנה מ-**4** עד **7** ו-**i - Cell_ N** - מ-**0** עד **3**;

כאשר 8 = **Cell_N**, אז i משתנה מ-**8** עד **11**, ו-**i - Cell_ N** - שוב מ-**0** עד **3** וכדומה

נותר רק להכניס לתחילת הפרוצדורה את מונה הלחיצות:

```
Click_Number = Click_Number + 1:
Private Sub cmd_Test_Click()
Click_Number = Click_Number + 1
If img_Dec(Cell_N).Tag = cmd_Code(0).Tag And _
        img_Dec(Cell_N + 1).Tag = cmd_Code(1).Tag And _
        img_Dec(Cell_N + 2).Tag = cmd_Code(2).Tag And _
        img_Dec(Cell_N + 3).Tag = cmd_Code(3).Tag Then
        For j = 0 To 3
          cmd_Code(j).Enabled = True
          cmd_Code(j).Picture =  cmd_Code(j).DownPicture
        Next
        img_Red.Visible = False
        img_Green.Visible = False
        img_Blue.Visible = False
        img_Yellow.Visible = False
        Frame1.Visible = True
        lbl_Message.Caption = "You have successfully solved it!"
Else
        If Click_Number = 4 Then
          Frame1.Visible = True
          lbl_Message.Caption = "You have no more attempts"
        End If
End If
For i = Cell_N To Cell_N + 3
If img_Dec(i).Tag = cmd_Code(i - Cell_N).Tag Then
        lbl_Order (i).ForeColor = vbRed
        lbl_Order (i).Caption = "P"
Else
        lbl_Order (i).Caption = "O"
End If
Next
Cell_N = Cell_N + 4
End sub
```

ננסה לנחש שוב את הקוד...

אמרנו כבר שקיים אלגוריתם מסוים לניחוש הקוד הסודי. עד שהאלגוריתם לא יימצא, יהיה עלינו להתחיל משחק מחדש כל פעם.

פרוצדורת התפריט **New Game** תיראה כך:

```
Private Sub mnu_New_Click()
```

1. בלולאה **i** מ-**0** עד **15** נסיר את התמונות מהמשבצות **img_Dec(i)**:

```
For i = 0 To 15
        img_Dec(i).Picture = LoadPicture()
```

2. נסיר את תגי המשבצות:

```
        img_Dec(i).Tag = " "
```

3. נסיר את תגי המשבצות **lbl_Order(i)** ונחזיר לאותיות את הצבע המקורי השחור:

```
        Order(i).Caption = " "
        Order(i).ForeColor = vbBlack
    Next
```

4. נוסיף קוד חדש ללחצנים **cmd_Code(j)**:

```
    For j = 0 To 3
        Randomize
        i = Int(4 * Rnd)
```

כדי לא לחזור כל פעם על שם הלחצן **cmd_Code(j)**, ניתן להשתמש במבנה **With ... End With**. מאפייני המרכיב, שעליו מדובר במבנה (במקרה שלנו - **cmd_Code(j)**) יירשמו כך:

```
            With cmd_Code(j)
                .DownPicture = LoadPicture(i)
                .Picture = LoadPicture()
                .Tag = i
                .Enabled = False
            End With
    Next
```

5. נאפס את המשתנים **Cell_N** ו-**Click_Number**.

```
        Cell_N = 0
        Click_Number = 0
```

6. נחזיר למרכיבים את מאפייני-המוצא שלהם:

```
        Frame1.Visible = False
        img_Red.Visible = True
        img_Green.Visible = True
        img_Blue.Visible = True
        img_Yellow.Visible = True
        cmd_Test.Enabled = False
End Sub
```

טקסט התוכנית המלא

כעת נבדוק האם התוכנית שלנו נכתבה נכון:

```
Dim i, j, Cell_N, Click_Number As Integer

Private Sub cmd_Test_Click()
Click_Number = Click_Number + 1
If img_Dec(Cell_N).Tag = cmd_Code(0).Tag And _
  img_Dec(Cell_N + 1).Tag = cmd_Code(1).Tag And _
  img_Dec(Cell_N + 2).Tag = cmd_Code(2).Tag And _
  img_Dec(Cell_N + 3).Tag = cmd_Code(3).Tag Then
   For j = 0 To 3
    cmd_Code(j).Enabled = True
    cmd_Code(j).Picture = cmd_Code(j).DownPicture
   Next
    img_Red.Visible = False
    img_Green.Visible = False
    img_Blue.Visible = False
    img_Yellow.Visible = False
    Frame1.Visible = True
    lbl_Message.Caption = "You have successfully solved it!"
Else
    If Click_Number = 4 Then
       Frame1.Visible = True
       lbl_Message.Caption = "You have no more attempts"
    End If
End If
For i = Cell_N To Cell_N + 3
If img_Dec(i).Tag = cmd_Code(i - Cell_N).Tag Then
   lbl_Order(i).ForeColor = vbRed
   lbl_Order(i).Caption = "P"
```

```
Else
  lbl_Order(i).Caption = "O"
End If
Next
Cell_N = Cell_N + 4
End Sub
```

```
Private Sub Form_Load()
    For j = 0 To 3
        Randomize
        i = Int(4 * Rnd)
        cmd_Code(j).DownPicture = LoadPicture(i)
        cmd_Code(j).Tag = i
    Next
End Sub
```

```
Private Sub img_Dec_DragDrop(Index As Integer, Source As Control, _
  X As Single, Y As Single)
  If Source.Tag <> "" Then
    img_Dec(Index).Picture = Source.Picture
    img_Dec(Index).Tag = Source.Tag
    cmd_Test.Enabled = True
  End If
End Sub
```

```
Private Sub mnu_New_Click()
    For i = 0 To 15
        img_Dec(i).Picture = LoadPicture()
        img_Dec(i).Tag = ""
        lbl_Order(i).Caption = " "
        lbl_Order(i).ForeColor = vbBlack
    Next
    For j = 0 To 3
        Randomize
        i = Int(4 * Rnd)
        With cmd_Code(j)
          .DownPicture = LoadPicture(i)
          .Picture = LoadPicture()
          .Tag = i
          .Enabled = False
        End With
```

```
  Next
    Cell_N = 0
    Click_Number = 0
    Frame1.Visible = False
    img_Red.Visible = True
    img_Green.Visible = True
    img_Blue.Visible = True
    img_Yellow.Visible = True
    cmd_Test.Enabled = False
End Sub
```

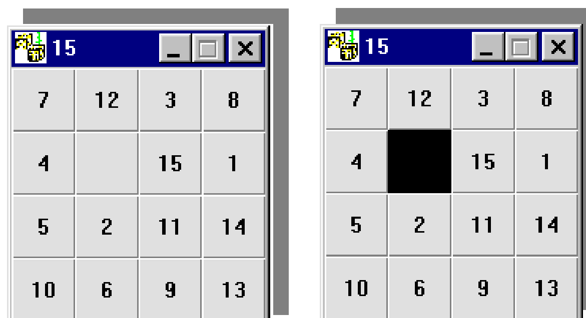
7 משחק "15"

כבר יותר מ-100 שנה משחק זה אינו מאבד את הפופולריות שלו ומוכר לכל ילד. כללי המשחק אינם מסובכים במיוחד: יש לסדר את האסימונים ברצף עולה, ממספר 1 עד מספר 15, ללא הוצאתם מהקופסה.

התוכנית הבסיסית של המשחק תכלול פרוצדורה אחת בלבד! אך למען האמת, בצורה זו המשחק יוצא פשוט מדי. אם נרצה לעשותו מורכב יותר, נוכל להוסיף מספר אפשרויות. בפרויקט זה נלמד גם כיצד אפשר לשמור תוצאת משחק.

יצירת ממשק המשחק

1. ניצור טופס שמאפייניו: **Width - 1992**, **Height - 2292**; **BackColor** - שחור.
2. נוסיף מערך **מ-16** הלחצנים (**CommandButton**) במידות **Width - 492**, **Height - 492**. ניצור את המערך בדיוק כפי שעשינו בפרויקט הקודם: נצייר מרכיב מערך ראשון בשם **Cmd_N** וניצור ממנו **15** עותקים.
3. נבדוק את שמות הלחצנים ואת אינדקס הלחצנים. אם המערך נוצר נכון, נקבל כתוצאה לחצנים מ-**cmd_N(0)** עד **cmd_N(15)**.



4. נכתוב (מאפיין **Caption**) מספרים על הלחצנים מ-**1** עד **15** בסדר מקרי, את אחד הלחצנים נשאיר ריק ונקבע לו מאפיין **Visible - False**. כשנפעיל את המשחק, נראה במקום לחצן זה את רקע הטופס.

5. נוסיף תמונה-Icon כלשהי לשורת הכותרת, נקבע לטופס **Caption - 15**. כדי שאי אפשר יהיה להגדיל את הטופס לגודל מלא נקבע למאפיין **MaxButton** - **False**. כך הלחצן האמצעי שמגדיל את הטופס, הופך ל"מת" (כמו **Enabled - False**).

כיצד להזיז אסימון

כדי להזיז אסימונים אין צורך להחליף את המקומות שלהם, די לגרום לכך שאסימון שעליו אנו לוחצים יהפוך לבלתי גלוי, ואסימון לא גלוי (הריבוע השחור הריק) יקבל משמעות **Visible** - **True**. מספר שנעלם מהלחצן (**Caption**) צריך לעבור ללחצן שהופיע כרגע, פעולה זו תיצור רושם של תנועת הלחצנים בתוך הטופס.

גירסה ראשונה (ופשוטה ביותר) של הפרוצדורה שלנו תיראה כך:

לסקור את הלחצנים **מ**-0 **עד** 15

אם לחצן שמספרו **i** בלתי גלוי, **אז**

להפוך את הלחצן הזה לגלוי

להעניק ללחצן הזה מספר של הלחצן, שלחצנו

להפוך את הלחצן שנלחץ לבלתי גלוי

סוף תנאי **"אם"**

לעבור ללחצן **הבא**

נתרגם זאת לשפת Visual Basic:

```
Private Sub cmd_N_Click(Index As Integer)
Dim i As Integer
For i = 0 To 15
 If cmd_N(i).Visible = False Then
   cmd_N(i).Visible = True
   cmd_N(i).Caption = cmd_N(Index).Caption
   cmd_N(Index).Visible = False
End If
Next
End Sub
```

נשים לב שהמשתנה **i** אינו מוגדר בחלק **General** כפי שעשינו קודם, אלא בתוך הפרוצדורה עצמה. אפשר לבצע זאת כך, אם המשתנה ישמש **רק בפרוצדורה זו** ולא במקום אחר.

כעיקרון, המשחק שלנו מוכן ואפשר לשחק בו, אם לא נעבור על כללי הזזת האסימונים. כלומר, לא ננסה להזיז אסימונים לריבוע ריק, כשהם אינם סמוכים אליו (ניתן ללחוץ למשל, על אסימון מספר **13**, והוא ידלג בהצלחה לריבוע הריק!). אך מוטב לקבוע מראש את התנאים האוסרים הזזות מסוג זה. למטרה זו יש לברר מהם המהלכים המותרים במשחק, ואז לקבוע הגבלות.

נסקור את סכימת מיקום האסימונים על הממשק לפי המספרים-אינדקסים שלהם. אנו כבר מכירים את מאפיין **Index** מהמשחק "בינגו", בפרויקט זה ניזכר כיצד להשתמש בו.

0	**1**	**2**	**3**
4	**5**	**6**	**7**
8	**9**	**10**	**11**
12	**13**	**14**	**15**

אם המערך נוצר נכון ולא התבלבלנו, סדר מיקום האינדקסים צריך להיות כמו בתמונה.

נניח, שקבענו את הלחצן עם אינדקס **5** כריבוע שחור ריק. אילו אסימונים ניתן להזיז למקומו?

מובן שמדובר בלחצנים **1**, **4**, **6** ו-**9**. אם למשל, הלחצן עם אינדקס 6 יהפוך לריבוע ריק, הזזות הלחצנים **2**, **5**, **7** ו-**10** תותרנה וכו'...

אם נסמן את אינדקס "הריבוע השחור" באות **i**, ניתן יהיה לקשר אותו לאינדקסי האסימונים המותרים להזזה ביחסים אלגבריים פשוטים: **i - 4**, **i - 1**, **i + 1** ו-**i + 4**.

נגדיר את מיקום "הריבוע השחור", שהרי הוא משתנה במשך המשחק כל הזמן ונגדיר את הגבלות ההזזות:

```
For i = 0 To 15
If cmd_N (i).Visible = False Then
   If i = Index + 1 Or _
      i = Index - 1 Or _
      i = Index + 4 Or _
      i = Index - 4 Then
          cmd_N (i).Visible = True
          cmd_N (i).Caption = cmd_N(Index).Caption
          cmd_N (Index).Visible = False
   End If
 End If
Next
```

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

כעת האסימונים נעים רק במגבלות המהלכים המותרים, ואפשר להציע לחברינו לשחק במשחק שלנו! כמובן, אם אין ביניהם נודניק סקרני במיוחד שירצה לבדוק קומבינציה זו או אחרת: למשל, כשהריבוע השחור נמצא במקום אסימון מספר **7**, יתחשק לו ללחוץ על אסימון מספר 8! במקרה זה כל החוקים שלנו מפסיקים לעבוד, והאסימון מקפץ בקלילות מעל לכל לוח המשחק! כדי למנוע תופעה בלתי רצויה זו, יש לשנות מעט את התוכנית:

```
Private Sub cmd_N_Click(Index As Integer)
Dim i As Integer
 For i = 0 To 15
  If cmd_N (i).Visible = False Then
   If i = Index + 1 And _
      Index <> 3 And i <> 4 And _
      Index <> 7 And i <> 8 And _
      Index <> 11 And i <> 12 Or _
      i = Index - 1 And _
      Index <> 4 And i <> 3 And _
      Index <> 8 And i <> 7 And _
      Index <> 12 And i <> 11 Or _
      i = Index + 4 Or _
      i = Index - 4 Then
          cmd_N (i).Visible = True
          cmd_N (i).Caption = cmd_N(Index).Caption
          cmd_N (Index).Visible = False
    End If
  End If
 Next
End Sub
```

כך אנו מונעים את הזזת האסימונים הבאים:

- אינדקס **3** אל מקום אינדקס **4** (ולהיפך),
- אינדקס **7** אל מקום אינדקס **8** (ולהיפך),
- אינדקס **11** אל מקום אינדקס **12** (ולהיפך).

אך יש פתרון יפה יותר שהוצע על ידי דניאל פפר בן ה-14:

```
Private Sub cmd_N_Click(Index As Integer)
Dim i As Integer
 For i = 0 To 15
  If cmd_N (i).Visible = False Then
   If cmd_N (Index).Left = cmd_N(i).Left Or _
     cmd_N (Index).Top = cmd_N(i).Top then
       If i = Index + 1 Or _
        i = Index - 1 Or _
        i = Index + 4 Or _
        i = Index - 4 Then
           cmd_N (i).Visible = True
           cmd_N (i).Caption = cmd_N(Index).Caption
           cmd_N (Index).Visible = False
     End if
   End If
  End If
 Next
End Sub
```

משמעות הפתרון היא שאסימון יכול לנוע רק מעלה-מטה, כלומר, רק אם הוא נמצא מעל או מתחת לריבוע השחור הריק. ניתן לרשום את התנאי כהשוואה בין הקואורדינטה השמאלית של האסימון לריבוע הריק:

```
cmd_N (Index).Left = cmd_N(i).Left
```

התנאי השני הקשור לאופרטור הלוגי הראשון "או", מאפשר הזזת האסימון ימינה-שמאלה. זאת אומרת, אם החלק העליון של האסימון (**Top**) נמצא באותה רמה של החלק העליון של הריבוע הריק: **cmd_N (Index).Top = cmd_N(i).Top**. מכאן, מובן שנאסרו קפיצות אסימונים לכיוונים אחרים.

כדי שמסגרת הפוקוס לא תופיע על האסימונים, נשתמש בשיטה המוכרת לנו זה מכבר מפרויקט מספר 2: ניצור בחלק הבלתי נראה של הטופס לחצן זעיר בשם **cmd_Focus** ונוסיף לתחילת הפרוצדורה את הפקודה:

```
cmd_Focus.SetFocus
```

כעת, תוכנית המשחק מוכנה באמת, והיא כוללת פרוצדורה אחת בלבד! מי שרוצה לגוון את המשחק יכול להוסיף מספר פרוצדורות.

האם ניתן לערבב אסימונים?

כאשר נשחק במשחק, באופן טבעי נרצה שהאסימונים ימוקמו אחרת בכל משחק חדש. דניאל פפר כתב את פרוצדורת ערבוב המספרים למשחק שלנו.

כדי שהפרוצדורה שלו תעבוד, צריך לשנות מספר תנאים.

1. נבחר את כל לחצני **cmd_N** ובחלון המאפיינים נמחק את המספרים שבמאפיין **Caption**.
2. הלחצן עם אינדקס 0 צריך להיות בלתי נראה (**Visible - False**), כל שאר הלחצנים - גלויים (**Visible - True**).
3. המשתנה-מונה **i** ישמש כעת בשתי הפרוצדורות, לכן נעביר את ההצהרה עליו לחלק **General**.
4. נכתוב את פרוצדורת טעינת הטופס:

```
Private Sub Form_Load()
Dim Number As Integer
   Number = 15
   Do
      Randomize
      i = Int((Rnd * 15) + 1)
      If cmd_N(i).Caption = "" Then
         cmd_N(i).Caption = Number
         Number = Number - 1
      End If
   Loop Until Number = 0
End Sub
```

כיצד עובדת פרוצדורה זו?

בפרוצדורה עצמה נגדיר את המשתנה **Number** כ-**Integer**, נקבע לו ערך תחילתי של **Number = 15**.

נארגן את עבודת הלולאה **... Do ... Loop Until**, לולאה זו מבצעת את כל הפקודות הבאות לאחר המילה **Do** ("עשה") וחוזרת עליהן (**Loop**) עד ש-(**Until**) יתבצע התנאי **Number = 0**.

בתוך גוף הלולאה מתבצעות הפקודות הבאות:

Randomize - ערבוב כל מספרי האסימונים;

i = Int((Rnd * 15) + 1) - בחירת מספר אקראי והקצאתו למשתנה **i**.

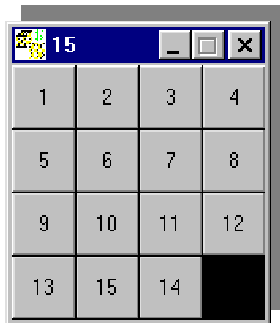
נבדוק האם קיים כיתוב כלשהו במאפיין **Caption** של האסימון שהאינדקס שלו תואם למספר האקראי **i**.

אם אין כיתוב, כלומר **" " = Caption**, נקצה למאפיין **Caption** את ערך המשתנה **Number**. מובן שהאסימון "הריק" הראשון יקבל מספר **15**. המיקום שבו יימצא האסימון תלוי בבחירה מקרית של האינדקס שלו (משתנה **i**).

נקטין את משתנה **Number** באחד. כעת האסימון הבא שמצבו בממשק המשחק תלוי במשתנה **i** יקבל ערך **14**.

הלולאה תחזור על עצמה עד שמשתנה **Number** יהיה שווה לאפס. את המספר הזה לא נקצה לשום אסימון, לכן עבודת הפרוצדורה הסתיימה. מעכשיו, בכל משחק חדש נקבל חלוקה אקראית של האסימונים.

הפרוצדורה עובדת, בכל משחק חדש אנו מקבלים חלוקת מספרים אקראית, אבל... לפעמים אי אפשר לסיים את המשחק! מדוע?



ננסה לשחק מספר פעמים. ייתכן שבסוף אחד מהמשחקים נגיע למיקום כזה של אסימונים. אל תבזבזו זמן על הזזות! בשנות השבעים של המאה הקודמת הובטח פרס 1000 דולר (סכום ענק בזמנו!) על פתרון בעיה זו. שיגעון המשחקים שהתפשט בכל אירופה ואמריקה לא נמשך זמן רב, המתמטיקאים קבעו שבעיה זו היא בלתי פתירה, אלא אם כן נוציא מהקופסה אסימונים (במשחק אמיתי ולא במחשב) ונחליף מיקומים.

אם כך, לצערנו לא נצטרך לפרוצדורת הערבוב היפה שכתב דניאל, שהרי היא עלולה להביא את המשחק למבוי סתום. אם כן, מה נעשה?

נשאיר את וריאנט מיקום המספרים הקודם, אחרת נצטרך לפנות לחישוב מתמטיקאי מסובך ולכתוב תוכנית שמגדירה מראש מצבים פתירים ולא פתירים.

כיצד להתחיל משחק חדש?

1. בעורך התפריט ניצור תפריט **New Game** בשם **mnu_New**.
2. נכתוב את הפרוצדורה לתפריט זה:

```
Private Sub nmu_New_Click()
   For i = 0 To 15
      cmd_N(i).Visible = True
   Next
   cmd_N(5).Visible = False

   cmd_N(0).Caption = "7"
   cmd_N(1).Caption = "12"
   cmd_N(2).Caption = "3"
   cmd_N(3).Caption = "8"
```

```
    cmd_N(4).Caption = "4"
    cmd_N(6).Caption = "15"
    cmd_N(7).Caption = "1"
    cmd_N(8).Caption = "5"
    cmd_N(9).Caption = "2"
    cmd_N(10).Caption = "11"
    cmd_N(11).Caption = "14"
    cmd_N(12).Caption = "10"
    cmd_N(13).Caption = "6"
    cmd_N(14).Caption = "9"
    cmd_N(15).Caption = "13"
    cmd_N(0).Caption = "7"
End Sub
```

בפרוצדורה לא מסובכת זו אנו הופכים את כל האסימונים לגלויים, מכיון שאיננו יודעים באיזה שלב נפסק המשחק הקודם, ואילו אסימונים נשארו בלתי נראים (**False - Visible**). לאסימון עם אינדקס 5 נקבע **False - Visible**, הוא יהיה האסימון "הריק" שלנו. לשאר האסימונים נקבע את המספרים הקודמים שלהם (**Caption**), משום שראינו כבר למה גורמת חלוקה מקרית של המספרים.

מונה מהלכים

אפשר לעשות את המשחק מעניין יותר אם נארגן בו "מונה מהלכים", במילים אחרות, נוסיף את המשתנה **Move_Counter**, שבו יצטבר מספר אחד עם כל לחיצה על הלחצן-אסימון. בתווית שנוסיף**lbl_Moves** יופיע מספר התואם למספר המהלך שבוצע. המנצח הוא זה שמסדר את כל האסימונים במינימום מהלכים.



1. נגדיל את הטופס שלנו כדי שיהיה מקום לתווית (**Label**) מונה המהלכים **lbl_Moves** ולתווית רגילה עם כיתוב **Moves**.
2. במאפיין **Caption** נקבע ערך 0 ונגדיר את גודל הגופן המתאים (מאפיין - **Font**).
3. במאפיין **Alignment** נקבע **Center**.
4. בחלק **General** נגדיר את המשתנה **Move_Conter** כ-**Integer**.
5. לתחילת הפרוצדורה של הלחצנים נוסיף שתי פקודות:

```
Move_Counter = Move_Counter + 1
lbl_Moves.Caption = Move_Counter
```

כעת, עם כל לחיצה על לחצן, יופיע בתווית **lbl_Moves** ערך המשתנה **Move_Counter** - מספר המהלכים שבוצעו במשך המשחק.

אפשר לכבות את מונה המהלכים. למטרה זו נוסיף תפריט **Options** ובו שני פריטים: **Count moves** בשם **itm_Count**, ו-**Show scores** בשם **itm_Scores** (לפריט זה נזדקק בהמשך הפרויקט). כעת, מונה המהלכים יעבוד רק אם נבחר בפריט **Count moves**.

השתמשנו כבר במאפיין **Checked**, כדי לסמן את פריט התפריט הנבחר. לכן, נפתח את תפריט **Options**, נלחץ על הפריט **Count moves** ונכתוב את הפרוצדורה הפשוטה:

```
Private Sub itm_Count_Click()
  itm_Count.Checked = True
End Sub
```

מונה המהלכים יעבוד בתנאי שפריט זה מסומן, **itm_Count.Checked = True**. כך נראית הפרוצדורה הבסיסית שלנו:

```
Private Sub cmd_N_Click(Index As Integer)
cmd_Focus.SetFocus
If itm_Count.Checked = True Then
    Moves_Counter = Moves_Counter + 1
    lbl_Moves.Caption = Moves_Counter
Else
    itm_Count.Enabled = False
End If
For i = 0 To 15
 If cmd_N(i).Visible = False Then
   If cmd_N(Index).Left = cmd_N(i).Left Or _
      cmd_N(Index).Top = cmd_N(i).Top Then
      If i = Index + 1 Or _
         i = Index - 1 Or _
         i = Index + 4 Or _
         i = Index - 4 Then
         cmd_N(i).Visible = True
         cmd_N(i).Caption = cmd_N(Index).Caption
         cmd_N(Index).Visible = False
      End If
    End If
  End If
Next
End Sub
```

הנה פרט מעניין: אם לא נדליק את מונה המהלכים לפני תחילת המשחק, הפריט **itm_Count** יהיה בלתי נגיש (**Enabled - False**) מייד לאחר הלחיצה הראשונה. כעת נראה אם מישהו יצליח "לעבוד" על המשחק וינסה להדליק את המונה בסוף המשחק!

כדי לאפס את מונה המהלכים בעת משחק חדש, נוסיף לפרוצדורה של **mnu_New** את הפקודות הבאות:

```
itm_Count.Checked = False
itm_Count.Enabled = True
Moves_Counter = 0
lbl_Moves.Caption = 0
```

שמירת תוצאת המשחק

Visual Basic מאפשרת לנו לרשום את תוצאת המשחק במקום ספציפי שנקרא **Windows Registry**. אך לפני שמירת התוצאה שהיא מספר המהלכים, עלינו לבדוק שכל האסימונים מסודרים כנדרש. בזה תעזור לנו פרוצדורה שכתב... דניאל פפר.

דניאל קרא לה **Check_if_Won** והגדיר אותה כפרוצדורה בפני עצמה שאינה קשורה לאף אחד מהמרכיבים, **Public Sub Check_If_Won()**. בתוך הפרוצדורה הוגדרו שלושה משתנים חדשים: **Number**, **Checker** ו-**Index**.

המשתנה **Number** הוא בעצם הכיתוב (**Caption**) שעל הלחצנים שלנו. עלינו להתאים בין המשתנה **Number** למשתנה **Index** אשר משקף את מיקום האסימון על לוח המשחק. השתמשנו כבר במשתנה **Index** בפרוצדורה הבסיסית, אך משום שמדובר במשתנה **לוקלי** (כלומר, משתנה המוגדר בתוך פרוצדורה אחת בלבד), כדאי להגדיר אותו שוב, אם משתמשים בו בפרוצדורה אחרת (למען האמת, Visual Basic יכולה גם "לסלוח" לנו אם "נשכח" להגדיר אותו שוב).

משתנה **Checker** הוא מונה שיפחת באחד בכל פעם שתתגלה התאמה בין מיקום ומספר האסימון.

```
Public Sub Check_if_Won()
Dim Number, Checker, Index As Integer
Number = 1
Checker = 15
Index = 0
Do Until Checker = 0
   If cmd_N(Index).Caption = Number Then
      Checker = Checker - 1
   Else
      Exit Do
   End If
 Index = Index + 1
 Number = Number + 1
Loop
If Checker = 0 Then
        Msg = MsgBox("Save score?", vbQuestion + vbYesNo, "Save score?")
End Sub
```

כיצד עובדת הפרוצדורה?

נקצה למשתנה **Checker** ערך **15**, למשתנה **Index - 0** ולמשתנה **Number - 1**.

נכתוב לולאה שבה תתבצע בדיקת התנאי **If cmd_N(Index).Caption = Number**, במילים אחרות, הצעד הראשון (איטרציה) של הלולאה בודק האם מספר (**Caption**), ששווה ל-**1** נמצא על הלחצן עם אינדקס **0** (נזכור שלחצן עם אינדקס זה הוא הראשון ב-**16** הלחצנים). אם התנאי מתקיים, המשתנה **Checker** מופחת באחד: **Checker = Checker - 1**. אם לא (**Else**), הלולאה נפסקת: **Exit Do**.

הלולאה **Do Until ... Loop** עובדת עד שהתנאי **Checker = 0** מתקיים.

כאשר כל הלחצנים נמצאים במקומם הלולאה מסתיימת, מופעל התנאי המוביל להופעת תיבת הודעה **Message Box** (**MsgBox**) עם ההודעה (**Msg**):

```
If Checker = 0 Then
Msg = MsgBox("Save score?", vbQuestion + vbYesNo, "Save score?")
```

בתיבת ההודעה מואר הכיתוב **"Save score?"**, מופיע סימן השאלה - **vbQuestion**, שני לחצני הפקודה **"Yes"** ו-**"No"** - **vbYesNo** ובשורת הכותרת של תיבת ההודעה גם מופיע הכיתוב **"Save score?"**.

כדי לבצע את הבדיקה יש להוסיף את הפרוצדורה לסוף פרוצדורת המשחק הבסיסית:

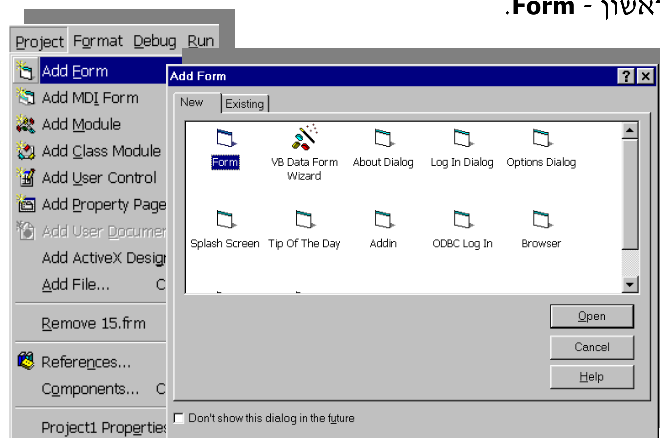
```
Private Sub cmd_N_Click(Index As Integer)
 cmd_Focus.SetFocus
 If itm_Count.Checked = True Then
    Moves_Counter = Moves_Counter + 1
    lbl_Moves.Caption = Moves_Counter
 Else
    itm_Count.Enabled = False
 End If
 For i = 0 To 15
  If cmd_N(i).Visible = False Then
    If cmd_N(Index).Left = cmd_N(i).Left Or _
       cmd_N(Index).Top = cmd_N(i).Top Then
       If i = Index + 1 Or _
          i = Index - 1 Or _
          i = Index + 4 Or _
          i = Index - 4 Then
          cmd_N(i).Visible = True
          cmd_N(i).Caption = cmd_N(Index).Caption
          cmd_N(Index).Visible = False
       End If
     End If
   End If
 Next
Check_if_Won                 'פרוצדורת הבדיקה
End Sub
```

מעתה כל הזזת אסימונים תלוית בבדיקה, וכאשר כל האסימונים יהיו במקומם, תופיע תיבת ההודעה הבאה:



הצעד הראשון שלנו יהיה תוכנית רישום תוצאות המשחק ב- **Windows Registry**. לשם כך נשתמש בטופס נפרד.

1. בתפריט **Project** שבשורת התפריטים של Visual Basic נבחר **Add Form** ואחר את הטופס הראשון - **Form**.



2. נמקם על הטופס תיבת טקסט שתשמש להקלדת שם השחקן - **txt_Name**. מעליה נמקם תיבת טקסט יותר שתשמש להצגת התוצאה - **txt_Score** ונוסיף להן תוויות מתאימות (**Labels**): "**Enter your name**" ו-"**Score**".

3. לתווית **Label2** עם הכיתוב "**Score**" ולתיבת הטקסט **txt_Score** נקצה ערך **False - Visible**.

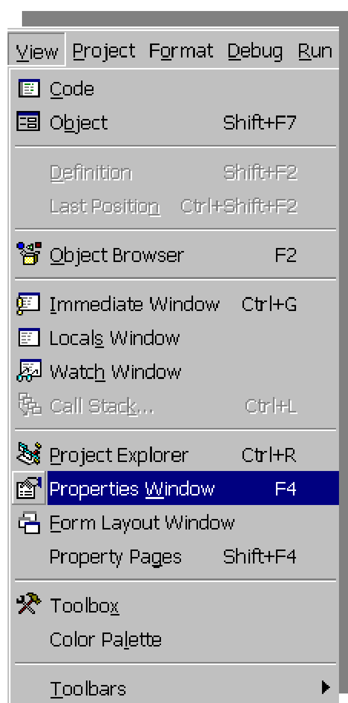
4. נציב על הטופס תיבת הסימון - **Check Box**, ונקצה למאפיין **Caption** את הכיתוב **Save score**. למאפיין **Value** נקצה ערך **Unchecked - 0**.

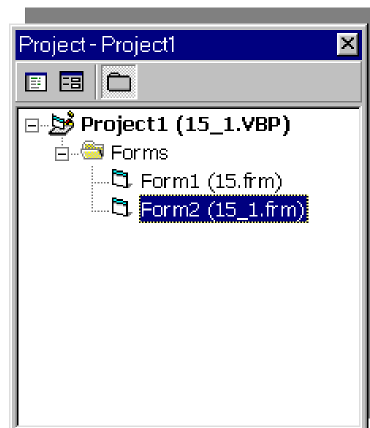
5. נוסיף לחצן פקודה רגיל **OK** בשם **cmd_OK**.

6. בעורך התפריטים ניצור שני תפריטים: **Show last scores** בשם **mnu_Show** ו-**Exit** בשם **mnu_Exit**. בתפריט **Show last scores** נסיר את הסימון ליד האפשרות **Enabled**, כלומר נהפוך אותו **False - Enabled**.

7. במאפיין **Caption** של הטופס נכתוב **Scores** ונקבע את מאפיין **Control Box** במצב **False**, פעולה זו תסיר את הלחצן ☒ שבפינה הימנית עליונה של הטופס.

כעת, יש לארגן את קריאת טופס מספר 2 מטופס מספר 1. לשם כך, נחזור לפרוצדורות טופס מספר 1. ניתן לעשות זאת באמצעות **Project Explorer** שנמצא בתפריט **View** של שורת התפריטים וגם באמצעות סרגל הכלים.





8. בחלון **Project** נבחר את הטופס הראשון **Form1** ונעבור לחלון הקוד.

9. נפתח את הפרוצדורה **Sub Check_if_Won()**. אנו צריכים לקרוא לטופס 2 מתיבת ההודעה (**Message Box**), כאשר היא מציעה לשמור את תוצאות המשחק:

```
Sub Check_if_Won()
Dim Number, Checker, Index As Integer
Number = 1
Checker = 15
Index = 0
Do Until Checker = 0
   If cmd_N(Index).Caption = Number Then
      Checker = Checker - 1
   Else
      Exit Do
   End If
 Index = Index + 1
 Number = Number + 1
Loop
If Checker = 0 Then
Msg = MsgBox("Save score?", vbQuestion + vbYesNo, "Save score?")
       If Msg = vbYes Then
         Form2.Show
         With Form2
           .Label1.Caption = "Enter your name"
                 .txt_Name.Text = ""
                 .txt_Score.Text = ""
                 .chk_Save.Visible = True
                 .chk_Save.Value = Unchecked
                 .cmd_OK.Visible = True
          End With
        Else
         End
        End If
End Sub
```

קריאת טופס מספר 2 מתבצעת באמצעות הפקודה **Form2.Show**, שתתבצע בתנאי שנלחץ על הלחצן "**Yes**" (**vbYes**). אחרת (**Else**), תתבצע יציאה מתיבת ההודעה (**End**).

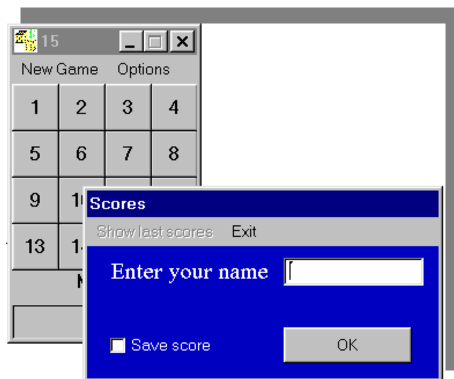
בהמשך תופיע סדרת הפקודות שתינתנה למרכיבים הנמצאים על טופס מספר 2. מכיון שמרכיבים אלה מנוהלים מטופס מספר 1, עלינו לציין כל פעם את הכתובת שלהם, במילים אחרות, להראות לתוכנית את הדרך למרכיב, למשל:
Form2.txt_Name.Text = " " או: **Form2.txt_Score.Text = " "**.

רישום כזה מאריך את התוכנית, ול-Visual Basic יש שיטה נוחה לקצר אותו. נכתוב **With Form2**, אחר נרשום את כל מה שקשור למרכיבי טופס 2 ונסגור את הקטע בפקודה **End With**:

```
With Form2
    .Label1.Caption = "Enter your name"
    .txt_Name.Text = ""
    .txt_Score.Text = ""
    .chk_Save.Visible = True
    .chk_Save.Value = Unchecked
    .cmd_OK.Visible = True
End With
```

כעת, אם נסדר נכון את כל האסימונים מ-**1** עד **15**, נקבל את תיבת ההודעה, נענה "**Yes**" וכך נציג על המסך את טופס מספר 2.

1. בעזרת **Project Explorer** נעבור לטופס מספר 2 ונפתח את חלון הקוד.



2. להלן הפרוצדורה ללחצן **cmd_OK**:

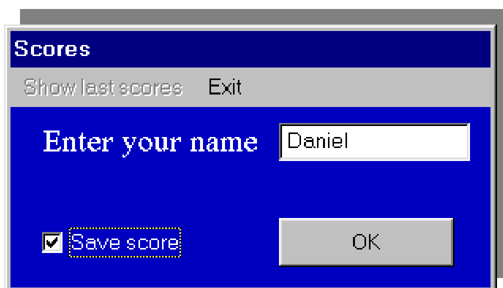
```
Private Sub cmd_OK_Click()
  If chk_Save.Value = Checked Then
   SaveSetting appname:="15",_
            section:="Score", _
            Key:=txt_Name.Text, _
            Setting:=Form1.lbl_Moves.Caption
  End If
  mnu_Show.Enabled = True
  Label1.Caption = "Player's name"
  Label2.Visible = True
  txt_Score.Visible = True
  cmd_OK.Visible = False
  chk_Save.Visible = False
End Sub
```

זו הפרוצדורה הראשונה שבה השתמשנו בפונקציה של Visual Basic שנקראת **SaveSettings**. פונקציה זו מחייבת הגדרת ארבעה ארגומנטים:

- **appname** - שם היישום (תוכנית). שם זה מוקצה לתיקייה (**Folder**) שבה תישמרנה התוצאות. נקרא לתיקייה "15", לפי שם המשחק.
- **section** - תיקיית המשנה **Scores**, שבה תישמרנה תוצאות הרישום שלנו.
- **key** - הרישום הנשמר. במקרה שלנו רישום זה תואם למאפיין טקסט (**Text**) של תיבת הטקסט **txt_Name**.
- **setting** - התוצאה הנשמרת, כלומר מספר התנועות **Moves_Counter** שעלינו לקחת מהטופס הראשון (טופס המשחק) **Form1**, מתווית **lbl_Moves**. כדי למצוא את התוצאה, יש לציין קודם את הטופס, נקודה ולאחריה את המרכיב והמאפיין שלו התואם למשתנה **Moves_Counter**, כך:

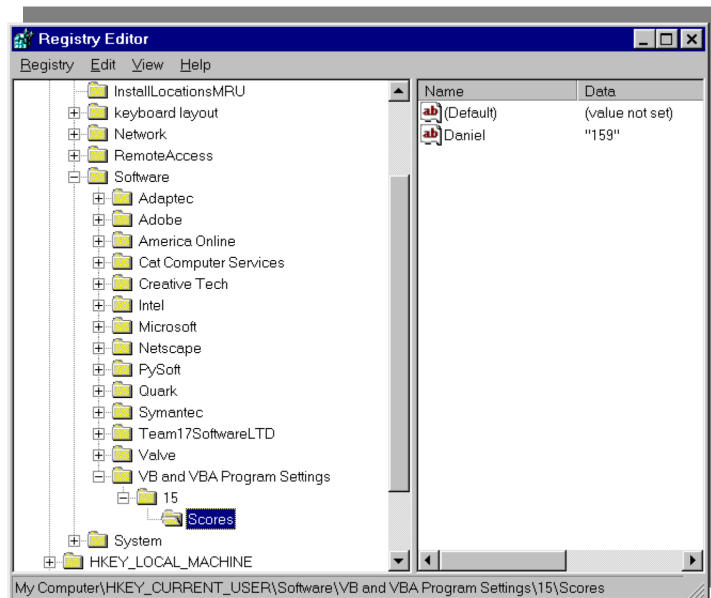
```
Setting:= Form1.lbl_Moves.Caption
```

נראה לאן תירשמנה התוצאות.



1. נקליד את שם השחקן לתיבת הטקסט.
2. נציב את הסימן ✓ ליד "**Save Score**" במצב מופעל.
3. נלחץ על הלחצן **OK**.
4. כעת יש להפעיל את **Regedit.exe** שנמצא בתיקיית Windows.
5. ב- **Regedit.exe**, בתפריט **Edit** נבחר בפריט **Find** (מצא) ונחפש את שם השחקן המוקלד. לאחר מספר שניות, התוכנית תפתח את התיקייה הנקראת 15, שבה נמצאת התיקייה **Scores**, ובתוכה ימצא הרישום **Daniel**. אם לא השתמשנו במשחק באפשרות **Count Moves** התוצאה (**Score**) תהייה 0. אם כן השתמשנו, התוצאה תהייה שווה למספר המהלכים.

להלן חלון Registry Editor :



כדי שהשחקן יוכל לראות את התוצאה הנשמרת האחרונה שלו, הכנסנו לתפריט **Options** של תפריט טופס מספר 1 את הפריט **Show scores**. נצמיד לו את הפרוצדורה הבאה :

```
Private Sub itm_scores_Click()
    Form2.Show
    With Form2
        .Label1.Caption = "Enter your name"
        .txt_Name.Text = ""
        .txt_Score.Text = ""
        .chk_Save.Visible = False
        .chk_Save.Value = Unchecked
        .chk_Save.Visible = False
        .cmd_OK.Visible = True
      End With
End Sub
```

פרוצדורה זו דומה במקצת לפרוצדורת הלחצן **vbYes** שבתיבת ההודעה **Message Box**, היא מציגה על המסך את טופס מספר 2, וכך הוא נראה:

Scores
Show last scores Exit
Player's name Daniel
Score

Scores
Show last scores Exit
Enter your name
OK

כאשר אנו מקלידים שם שיש לו רישום ב- **Windows Registry**, מופעל תפריט **Show last scores** ומופיעים הכיתוב **Players name** ותיבת הטקסט **txt_Score** ובה תוצאת המשחק האחרון.

Scores
Show last scores Exit
Player's name Daniel
Score 159

כדי שהתוכנית תעבוד, יש לכתוב את הפרוצדורה לתפריט **Show last scores**:

```
Private Sub mnu_Show_Click()
    txt_Score.Text = GetSetting(appname:="15",_
                section:="Scores", _
                Key:=txt_Name.Text, _
                Default:="")
End Sub
```

בפרוצדורה זו אנו משתמשים בפונקציה **GetSetting**, שמכניסה לתיבת הטקסט **txt_Score** את הטקסט התואם לרישום הבסיסי (**Default**). הארגומנטים של הפונקציה **GetSetting** (**appname**, **section** ו-**key**) הם אותם הארגומנטים של הפונקציה **SaveSetting**.

ליציאה מטופס מספר 2 אל טופס מספר 1 (טופס המשחק הבסיסי) יש להשתמש בתפריט **Exit**:

```
Private Sub mnu_Exit_Click()
    Form2.Hide
End Sub
```



נסיים בהערה הקטנה: אם ננסה ללחוץ על הלחצן **OK** מבלי להקליד שם בתיבת הטקסט **txt_Name**, התוכנית תודיע על שגיאה ("באג"). כדי להימנע מכך, נקצה ללחצן **cmd_OK** במאפיין **Enabled** - **False**, ולתיבת הטקסט **txt_Name** נוסיף את הפרוצדורה הבאה:

```
Private Sub txt_Name_Change()
  cmd_OK.Enabled = True
End Sub
```

מעתה, כל סימן שיוקלד לתיבת הטקסט, יגרום להתרחשות האירוע **txt_Name_Change()**, וכתוצאה מכך, לביצוע הפקודה **cmd_OK.Enabled = True**.

סיימנו את הפרויקט. כעת יש לבדוק תוך כדי משחק, האם נשארו פרוצדורות כלשהן שאינן גמורות, או אם יש אולי שגיאות בתוכנית. לכן, נשווה את הטקסטים של הפרוצדורות שלנו עם טקסט התוכנית המלא.

הטקסט המלא של תוכנית "15"

טופס מס' 1

```
Dim i, Moves_Counter As Integer

Private Sub cmd_N_Click(Index As Integer)
 cmd_Focus.SetFocus
 If itm_Count.Checked = True Then
     Moves_Counter = Moves_Counter + 1
     lbl_Moves.Caption = Moves_Counter
 Else
     itm_Count.Enabled = False
 End If
 For i = 0 To 15
  If cmd_N(i).Visible = False Then
    If cmd_N(Index).Left = cmd_N(i).Left Or _
      cmd_N(Index).Top = cmd_N(i).Top Then
        If i = Index + 1 Or _
          i = Index - 1 Or _
          i = Index + 4 Or _
          i = Index - 4 Then
         cmd_N(i).Visible = True
         cmd_N(i).Caption = cmd_N(Index).Caption
         cmd_N(Index).Visible = False
       End If
    End If
  End If
 Next
Check_if_Won
End Sub

Sub Check_if_Won()
Dim Number, Checker, Index As Integer
Number = 1
Checker = 15
Index = 0
Do Until Checker = 0
    If cmd_N(Index).Caption = Number Then
        Checker = Checker - 1
    Else
        Exit Do
    End If
```

```
 Index = Index + 1
 Number = Number + 1
Loop
If Checker = 0 Then
   Msg = MsgBox("Save score?", vbQuestion + vbYesNo, "Save score?")
          If Msg = vbYes Then
            Form2.Show
            With Form2
              .Label1.Caption = "Enter your name"
              .txt_Name.Text = ""
              .txt_Score.Text = ""
               .chk_Save.Visible = True
               .chk_Save.Value = Unchecked
               .cmd_OK.Visible = True
            End With
         Else
            End
         End If
End If
End Sub

Private Sub itm_Count_Click()
   itm_Count.Checked = True
End Sub

Private Sub itm_Scores_Click()
    Form2.Show
    With Form2
        .Label1.Caption = "Enter your name"
        .txt_Name.Text = ""
        .txt_Score.Text = ""
        .chk_Save.Visible = False
        .chk_Save.Value = Unchecked
        .chk_Save.Visible = False
        .cmd_OK.Visible = True
    End With
End Sub
Private Sub mnu_New_Click()
    For i = 0 To 15
        cmd_N(i).Visible = True
    Next
```

```
    cmd_N(5).Visible = False
    cmd_N(0).Caption = "7"
    cmd_N(1).Caption = "12"
    cmd_N(2).Caption = "3"
    cmd_N(3).Caption = "8"
    cmd_N(4).Caption = "4"
    cmd_N(6).Caption = "15"
    cmd_N(7).Caption = "1"
    cmd_N(8).Caption = "5"
    cmd_N(9).Caption = "2"
    cmd_N(10).Caption = "11"
    cmd_N(11).Caption = "14"
    cmd_N(12).Caption = "10"
    cmd_N(13).Caption = "6"
    cmd_N(14).Caption = "9"
    cmd_N(15).Caption = "13"
    itm_Count.Checked = False
    itm_Count.Enabled = True
    Moves_Counter = 0
    lbl_Moves.Caption = 0
End Sub
```

8 משחק "זיכרון 2"

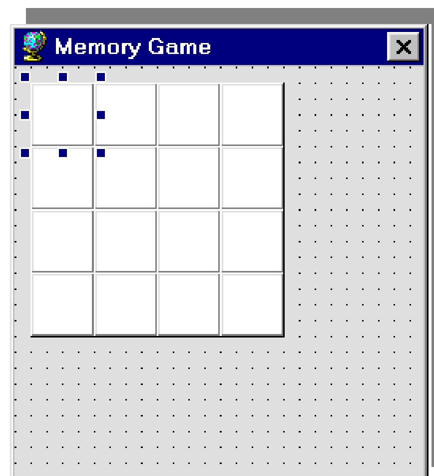
בפרק 2 בנינו משחק דומה. כעת, משרכשנו ניסיון כלשהו בתכנות ב- Visual Basic, נוכל לבנות את תוכנית המשחק בצורה מעניינת יותר.

תיקיית הקבצים

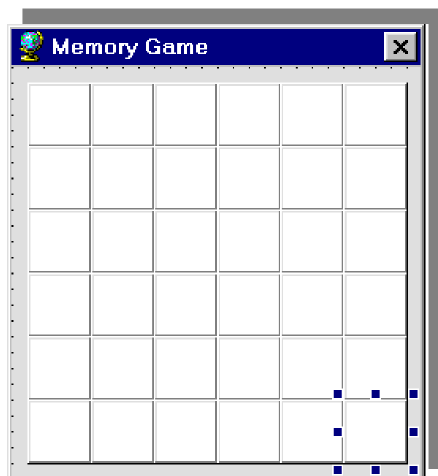
לשמירת קבצי הפרויקט ניצור תיקיה מיוחדת בשם **Memory Game**, אליה נעתיק את קבצי התמונות **0-49** מתיקיית הפרויקט "בינגו". הקבצים נמצאים גם בתקליטור, המצורף לספר, בתיקיה VB Projects\Part 8. לתיקיה החדשה נעתיק גם את הקובץ **card.bmp**.

יצירת ממשק המשחק

1. נפתח פרויקט חדש ונציב על הטופס מערך מ-16 לחצנים בשמות **cmd_Pict(0)...cmd_Pict(15)** בגדלים **Width - 492**, **Height - 492**, כמו בתמונה. את המאפיין **Style** נקבע כ-**Graphical** ונצבע את הלחצנים בצבע לבן. נקבע למאפיין **Visible** את הערך **False**.
2. לאחר מכן, נוסיף למערך עוד 20 לחצנים ונמקם אותם בצורת שני טורים מימין ושתי שורות למטה, כדי שיתקבל מערך מ-36 לחצנים.
3. נקבע את גובה ורוחב הטופס. כאשר נמצאים עליו **16** לחצנים, נקבע למשל את הגדלים: **Width - 2300**, **Height - 2950**, וכאשר יהיו **36** לחצנים, נקבע למשל **Width - 3180**, **Height - 3760**.



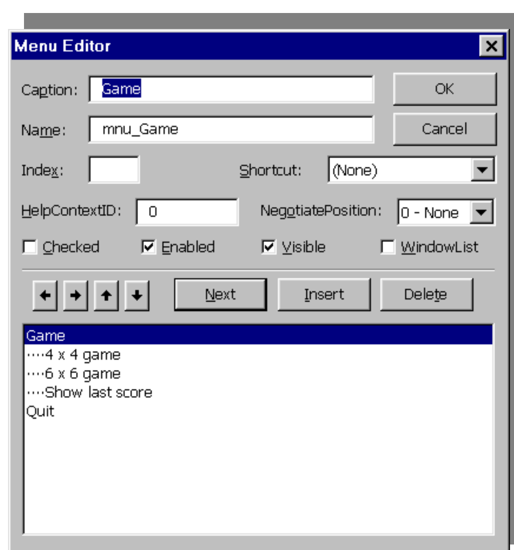
מערך מ-16 לחצנים



מערך מ-36 לחצנים

4. יוצרים תפריט לפי הטבלה והתמונה:

Game		mnu_Game
	4 x 4 game	itm_4
	6 x 6 game	itm_6
	Show last score	itm_Scores
Quit		mnu_Quit



הגדרת משתנים

לחלק **General** נכניס שורה נוספת **Option Explicit** ונגדיר את המשתנים הבאים:

```
Option Explicit
Dim Place, ImageN, Number, Test, i, LostIndex As Integer
Dim Msg
Public ClickNumber As Integer
```

בשונה משיטת הגדרת המשתנים המוכרת לנו עד כה - הגדרה באמצעות מילת המפתח **Dim**, כאן אנו משתמשים לראשונה במילה **Public**. מדוע? נדון בכך מאוחר יותר.

שים לב להגדרת המשתנה **Msg**. אם לא נקבע למשתנה סוג מסוים, Visual Basic תקצה לו **כברירת מחדל** את הסוג **Variant**, שיכול לקבל כמעט כל ערך. נוח, נכון? ונראה שאין אנו חייבים להגדיר סוג מסוים בכל הגדרת משתנה - די לכתוב את שמו! יותר מזה, אפשר גם שלא להגדיר משתנה בתחילת התוכנית, אלא פשוט להשתמש בו בפרוצדורות ללא הגדרה מראש.

זה נכון שבשונה משפות אחרות, Visual Basic היא שפה גמישה מאוד. למשל ב-Pascal, התנהגות חופשית כזו אסורה לחלוטין. אך בגישה זו טמונים סיכונים מסוימים, למשל סיכוי לשגיאה, או שימוש לא נכון במשתנים. כדי להבטיח עצמנו מבעיות אלו, אפשר לקבוע בחלק **General** את הפקודה המיוחדת **Option Explicit**, המיועדת לבדיקת שמות המשתנים בתוכנית. אם נסיר פקודה זו, אפשר יהיה להסתדר בלי הגדרת **Dim**, **Msg**, או משתנים אחרים כלשהם.

למרות ש-Visual Basic היא שפה גמישה מאוד, לא כדאי לנצל את אפשרות השימוש בהגדרות משתנים מקוצרות, או להימנע מהגדרות בכלל. שימוש ברישום מקוצר מסוג **Dim A** חוסך רק את זמן המתכנת ולא את המקום בזיכרון המחשב: הגדרת המשתנה מסוג **Variant** דורשת הרבה יותר מקום בזיכרון, מהגדרת משתנה מסוג **Integer**.

בחירת רמת המשחק

במשחק תהיינה שתי רמות קושי: **16** תמונות **(4 x 4)** ו-**36** תמונות **(6 x 6)**. אפשר גם לבנות רמות קושי נוספות, אך לשם כך יש לדאוג למיקום נכון של הלחצנים.

1. לרמה **4 x 4** נכתוב את הפרוצדורה הבאה:

```
Private Sub itm_4_Click()
```

2. נקבע למשתנה **Test** ערך התחלתי השווה למספר הלחצנים - **16**:

```
        Test = 16
```

3. נציב סימן ✓ ברמה הנבחרת, ונסיר אותו מהרמה השנייה (אם הוצב שם קודם):

```
        itm_4.Checked = True
        itm_6.Checked = False
```

4. נקבע את גודל הטופס בהתאם למספר הלחצנים שעליו:

```
Form1.Width = 2300
Form1.Height = 2950
```

5. ניצור לולאה ונקבע ללחצנים שישתתפו במשחק את המאפיין **True - Visible**. הקובץ **card.bmp** מוצב בלחצנים כ"מכסה", ואם נשארו תמונות מהמשחק הקודם יש להסירן.

```
For i = 0 To 15
  cmd_Pict(i).Visible = True
  cmd_Pict(i).Picture = LoadPicture("card.bmp")
  cmd_Pict(i).DownPicture = LoadPicture()
Next
```

6. נערבב את מספרי התמונות ו"נוציא" מספר אקראי (**Number**):

```
Randomize
Number = Int(Rnd * 49)
```

7. כעת יש להכניס תמונות ללחצנים. לשם כך, נשתמש ברעיון הבא. יש לנו **16** לחצנים, כלומר, די לבחור שמונה תמונות ולהכניס אותן פעמיים כל אחת. מספרי הקבצים - תמונות יהיו בסדר רץ, אך המספר הראשון, שממנו מתחילה הסדרה יהיה מספר אקראי - **Number**. משום שאנו משתמשים בערבוב (**Randomize**) של מספרי הקבצים, "שרשרת" הקבצים תהיה שונה כל פעם. למשל, אם קיבלנו **17** כערך המשתנה **Number**, ממספר זה תחל הכנסת התמונות בסדר יורד: **17**, **16**, **15**, **14**, **13**, **12**, **11** ו-**10**. אפשר גם להפוך כיוון: להכניס קבצים בסדר עולה, אך לא זאת הבעיה. עלינו להגדיר תנאי הבודק אם המספר הראשון קטן מ-**8**, ואם כן, נוסיף לו **8** כדי שאפשר יהיה לספור שמונה מספרים "מטה":

```
  If Number < 8 Then Number = Number + 8
```

8. כדי שהפרוצדורה תחזור על עצמה פעמיים, ניצור לולאה:

```
  For i = 1 To 2
```

9. נקצה למשתנה **ImageN** ערך של מספר אקראי:

```
  ImageN = Number
```

10. נחפש לחצן פנוי ובו נכניס תמונה עם מספר **ImageN**, ניצור לולאה נוספת:

```
Do
```

11. נערבב את מספרי הלחצנים ונבחר מספר אחד אקראי:

```
        Randomize
Place = Int(Rnd * 16)
```

12. אם הלחצן עם המספר הזה פנוי (ללא תמונה), נכניס בו את התמונה (**ImageN**) הראשונה ונקצה לו תג (Tag) השווה למספר זה. אחר נקטין את המספר באחד:

```
If cmd_Pict(Place).DownPicture = LoadPicture() Then
            cmd_Pict(Place).DownPicture = LoadPicture(ImageN)
            cmd_Pict(Place).Tag = ImageN
            ImageN = ImageN - 1
End If
```

13. נסיים את הלולאה כאשר מספר התמונה יהיה שווה לערך ההתחלתי של המשתנה **Number** פחות שמונה:

```
Loop Until ImageN = Number - 8
```

14. בסוף הפרוצדורה נסגור את הלולאה **For...Next**:

```
Next
End Sub
```

הפרוצדורה לרמה **6 x 6** שונה מפרוצדורה זו רק בערכים המספריים של המשתנים:

```
Private Sub itm_6_Click()
         Test = 36
         itm_6.Checked = True
         itm_4.Checked = False
         Form1.Width = 3180
         Form1.Height = 3760
         For i = 0 To 35
           cmd_Pict(i).Visible = True
           cmd_Pict(i).DownPicture = LoadPicture()
           cmd_Pict(i).Picture = LoadPicture("card.bmp")
         Next
         Randomize
         Number = Int(Rnd * 49)
         If Number < 18 Then Number = Number + 18
           For i = 1 To 2
                     ImageN = Number
                     Do
                       Randomize
                       Place = Int(Rnd * 36)
                       If cmd_Pict(Place).DownPicture  = LoadPicture() Then
                          cmd_Pict(Place).DownPicture =  LoadPicture(ImageN)
                          cmd_Pict(Place).Tag = ImageN
                          ImageN = ImageN - 1
                       End If
                     Loop Until ImageN = Number - 18
           Next
End Sub
```

פתיחת התמונות

1. הנה הפרוצדורה הפשוטה לפתיחת התמונות - היא מעבירה את התמונה ממצב לא פעיל (מאפיין **DownPicture**) אל הלחצן עצמו (מאפיין **Picture**):

```
Private Sub cmd_Pict_Click(Index As Integer)
        cmd_Pict(Index).Picture = cmd_Pict(Index).DownPicture
 End Sub
```

2. כעת אפשר ליהנות מהמשחק: נלחץ על לחצן - תופיע תמונה, נלחץ על לחצן אחר תופיע תמונה אחרת וכו'. אך עדיין יש צורך בתיקונים. במשחק אמיתי, לאחר שלוחצים על התמונה השנייה התמונה הראשונה נסגרת, אלא אם כן התמונה השנייה זהה לראשונה, במקרה זה שתיהן נעלמות מהמסך. הבעיה היא שכאשר אנו לוחצים על הלחצן השני המחשב "שוכח" את האינדקס של הלחצן הראשון. לכן, המשתנה **LostIndex** צריך לשמור על אינדקס זה ברגע שהלחצן הראשון **מאבד** את הפוקוס. זה מתרחש בזמן שאנו מעבירים את העכבר ללחצן השני ולוחצים עליו. האירוע נקרא **LostFocus**:

```
Private Sub cmd_Pict_LostFocus(Index As Integer)
   LostIndex = cmd_Pict(Index).Index
End Sub
```

3. כעת אפשר להשוות את תגי הלחצנים. אם הם שווים, הלחצן הראשון והשני אמורים להיעלם, והמשתנה המונה את הלחצנים אמור לפחות ב-2. נחזור לפרוצדורה **Private Sub cmd_Pict_Click**:

```
   If cmd_Pict(LostIndex).Tag = cmd_Pict(Index).Tag Then
      cmd_Pict(LostIndex).Visible = False
      cmd_Pict(Index).Visible = False
      Test = Test - 2
   End If
```

4. יש עוד בעיה קטנה: הלחצן הראשון עם אינדקס **0** נעלם מהמסך כאשר אנו לוחצים עליו בעכבר. כדי למנוע זאת, נשנה את ערך המשתנה **LostIndex** בפרוצדורת טעינת הטופס:

```
Private Sub Form_Load()
   LostIndex = 1
End Sub
```

5. נשאר רק לארגן שלאחר לחיצה על הלחצן השני, הלחצן הראשון "ייסגר". זה יקרה כשנזיז את העכבר ללחצן השני ונלחץ עליו. הלחצן השני **מקבל** את הפוקוס ואירוע זה נקרא **GotFocus**. הפרוצדורה בודקת איזו תמונה נמצאת על הלחצן הראשון, ואם זה אינו קובץ **card.bmp**, היא תכניס קובץ זה ללחצן הראשון.

```
Private Sub cmd_pict_GotFocus(Index As Integer)
 If cmd_Pict(LostIndex).Picture <> LoadPicture("card.bmp") Then
       cmd_Pict(LostIndex).Picture = LoadPicture("card.bmp")
 End If
End Sub
```

המשחק שלנו מוכן! כעת אפשר לשחק בו ברמה **4 x 4** או ברמה **6 x 6**, לפי בחירה.

ספירת מהלכים

כמו שבמשחק "15", נוכל לספור בכמה מהלכים הסתיים המשחק. בכל לחיצה על תמונה יגדל המשתנה **ClickNumber** באחד, ותוצאת הפעולה של המונה תשתקף בכותרת הטופס. נוסיף לפרוצדורה **Private Sub cmd_Pict_Click** שתי פקודות:

```
ClickNumber = ClickNumber + 1
Form1.Caption = "It is your click # " & ClickNumber
```

התחלת משחק חדש

הפרוצדורה למשחק חדש פשוטה להפליא. היא תתבצע עם פתיחת תפריט **Game**. כדי שלא יישארו על הטופס לחצנים ממשחק קודם, ניצור לולאה שבה נסיר את הלחצנים מהטופס, נאפס את מונה המהלכים ונחזיר לטופס את הכותרת הקודמת שלו.

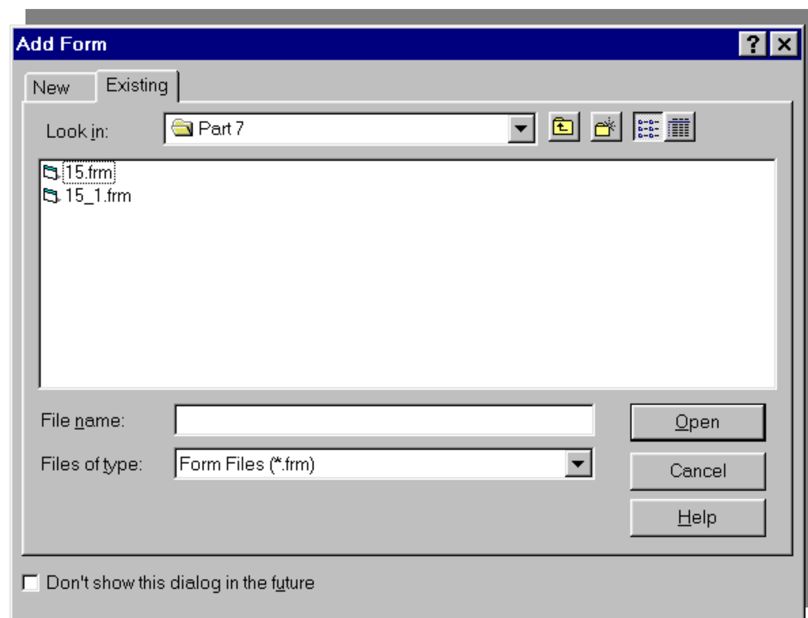
```
Private Sub mnu_Game_Click()
   For i = 0 To 35
       cmd_Pict(i).Visible = False
       cmd_Pict(i).DownPicture = LoadPicture()
   Next
   ClickNumber = 0
   Form1.Caption = "Memory Game"
End Sub
```

רישום התוצאה

נותר רק לרשום את תוצאת המשחק ב- **Windows Registry**, כפי שעשינו בפרויקט הקודם (משחק "15"). אין צורך לכתוב את הפרוצדורות מחדש, או לבנות את הטופס. נשתמש בטופס מהפרויקט הקודם. ניתן למצוא אותו גם בתקליטור בתיקיה VB Projects\Part 7. קובץ הטופס נקרא **15_1.frm**.

1. בחלק **Project** של שורת התפריטים, נבחר פריט **Add Form** ונפתח את כרטיסיית **Existing** - קיימים. נמצא את הטופס השני של הפרויקט "15".

2. נשמור את הטופס בשם אחר בתיקיית הפרויקט הנוכחי **Memory Game**.



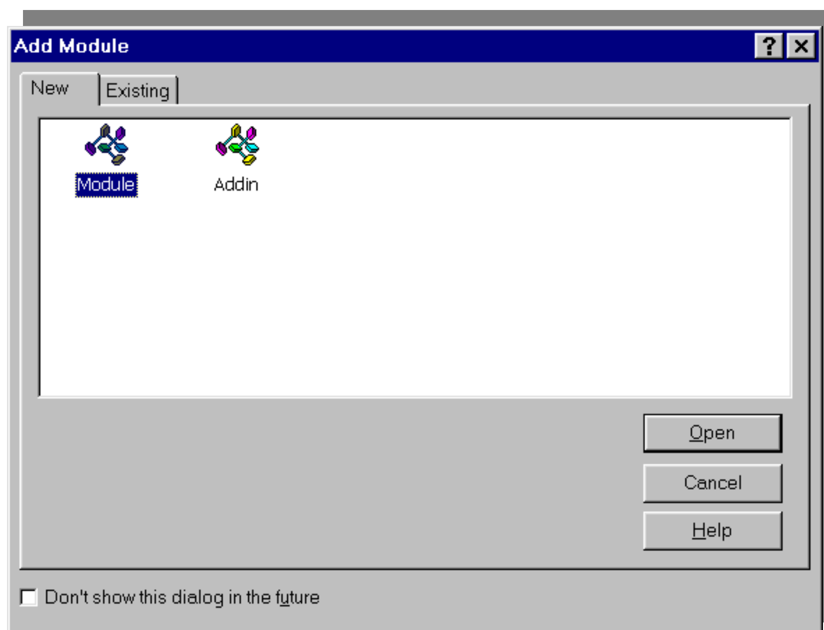
3. נוסיף שתי פקודות לסוף הפרוצדורה **Private Sub cmd_Pict_Click**:

```
If Test = 0 Then Msg = MsgBox("Save score?", _
  vbQuestion + vbYesNo, "Save score?")
If Msg = vbYes Then FormScore
```

הפקודה הראשונה (תנאי) מוכרת לנו מפרויקט "15". למעשה, העתקנו אותה משם, אך בשונה מהפרוצדורה שליווּתה אותו בפרויקט הקודם, כאן נכתוב פקודה מוזרה **FormScore**. מה המשמעות שלה? אל תנסה למצוא אותה ב-Help! **FormScore** הוא שם הפרוצדורה שיצרנו בעצמנו והפכנו אותה **למודול** נפרד. שוב, אין צורך ליצור אותה מחדש, היא נמצאת בפרויקט "15", אך היא מופיעה שם כחלק מפרוצדורה אחרת. מדובר בסדרת הפקודות שפותחות את הטופס השני **Form2** ומגדירות את שורת המאפיינים שלו.

```
Form2.Show
With Form2
    .Label1.Caption = "Enter your name"
    .txt_Name.Text = ""
    .txt_Score.Text = ""
    .chk_Save.Visible = True
    .chk_Save.Value = Unchecked
    .cmd_OK.Visible = True
End With
```

4. בחלק **Project** של שורת התפריטים, נבחר את הפריט **Add Module**:



5. נקרא לפרוצדורה **FormScore** ונגדיר אותה במודול:

```
Sub FormScore()
  Form2.Show
    With Form2
        .Label1.Caption = "Enter your name"
        .txt_Name.Text = ""
        .txt_Score.Text = ""
        .chk_Save.Visible = True
        .chk_Save.Value = Unchecked
        .cmd_OK.Visible = True
    End With
End Sub
```

6. נשמור את המודול בתיקיה שלנו בשם מסוים, למשל **ScoreMod**. לקבצים כאלה מוקצית באופן אוטומטי סיומת **bas**.

נוחות השימוש במודולים כאלה מובהקת:

- טקסט התוכנית הופך לקומפקטי יותר, קל לקריאה ואפשר לעקוב אחר היגיון התוכנית.
- סדרת פעולות שלמה שיכולה לשמש אחר כך בתוכניות אחרות, אפשר וצריך לשמור כמודולים נפרדים. דרך אגב, אם נשנה מעט את טקסט התוכנית, נוכל להשתמש במודול שלנו גם במשחק "15".

7. ניתן להשתמש בפרוצדורה שבמודול גם לפריט התפריט **Show last score**:

```
Private Sub itm_Scores_Click()
   FormScore
 End Sub
```

8. נעבור לחלון הקוד של טופס 2.

9. נשנה את הפרמטרים בפרוצדורה **SaveSetting**:

```
SaveSetting appname:="MemoryGame",_
            section:="Scores", _
            Key:=txt_Name.Text,_
            Setting:=Form1.ClickNumber
```

כאן נזקקנו למשתנה **ClickNumber** המוגדר על ידינו כ-**Public**. משתנים המוגדרים כ-**Public** משותפים לכל החלקים והמודולים של הפרויקט. למרות שמשתנה זה "עובד" בטופס הראשון, הוא "נראה" גם בטופס השני. לעומתם, משתנים המוגדרים באמצעות **Dim** הם לוקליים ו"נראים" רק בחלק פרויקט שבו הוגדרו.

נותר רק להשוות את מה שכתבנו עם טקסט הפרויקט המלא, והמשחק מוכן!

הטקסט המלא של התוכנית "זיכרון 2"

טופס מס' 1

```
Option Explicit
Dim Place, ImageN, Number, Test, i, LostIndex As Integer
Dim Msg
Public ClickNumber As Integer

Private Sub itm_4_Click()
   Test = 16
   itm_4.Checked = True
   itm_6.Checked = False
   Form1.Width = 2300
   Form1.Height = 2950
   For i = 0 To 15
    cmd_Pict(i).Visible = True
    cmd_Pict(i).DownPicture = LoadPicture()
    cmd_Pict(i).Picture = LoadPicture("card.bmp")
   Next
    Randomize
    Number = Int(Rnd * 49)
    If Number < 8 Then Number = Number + 8
    For i = 1 To 2
        ImageN = Number
        Do
            Randomize
            Place = Int(Rnd * 16)
            If  cmd_Pict(Place).DownPicture = LoadPicture() Then
                cmd_Pict(Place).DownPicture = LoadPicture(ImageN)
                cmd_Pict(Place).Tag = ImageN
               ImageN = ImageN - 1
            End If
         Loop Until ImageN = Number - 8
    Next
End Sub
```

```
Private Sub itm_6_Click()
  Test = 36
  itm_6.Checked = True
  itm_4.Checked = False
   Form1.Width = 3180
  Form1.Height = 3760
  For i = 0 To 35
   cmd_Pict(i).Visible = True
   cmd_Pict(i).DownPicture = LoadPicture()
   cmd_Pict(i).Picture = LoadPicture("card.bmp")
  Next
   Randomize
   Number = Int(Rnd * 49)
   If Number < 18 Then Number = Number + 18
   For i = 1 To 2
      ImageN = Number
      Do
         Randomize
         Place = Int(Rnd * 36)
         If  cmd_Pict(Place).DownPicture = LoadPicture() Then
            cmd_Pict(Place).DownPicture = LoadPicture(ImageN)
            cmd_Pict(Place).Tag = ImageN
            ImageN = ImageN - 1
         End If
      Loop Until ImageN = Number - 18
   Next
End Sub

Private Sub cmd_Pict_LostFocus(Index As Integer)
  LostIndex = cmd_Pict(Index).Index
End Sub

Private Sub cmd_pict_GotFocus(Index As Integer)
   If cmd_Pict(LostIndex).Picture <> _
     LoadPicture("card.bmp") Then
     cmd_Pict(LostIndex).Picture = _
     LoadPicture("card.bmp")
   End If
End Sub
```

```
Private Sub cmd_Pict_Click(Index As Integer)
   cmd_Pict(Index).Picture = cmd_Pict(Index).DownPicture
   If cmd_Pict(LostIndex).Tag = cmd_Pict(Index).Tag Then
       cmd_Pict(LostIndex).Visible = False
       cmd_Pict(Index).Visible = False
       Test = Test - 2
 End If
 ClickNumber = ClickNumber + 1
 Form1.Caption = "It is your click # " & ClickNumber
 If Test = 0 Then
    Msg = MsgBox("Save score?", vbQuestion + vbYesNo, "Save score?")
        If Msg = vbYes Then FormScore
  End If
End Sub
```

```
Private Sub Form_Load()
    LostIndex = 1
End Sub
```

```
Private Sub itm_Scores_Click()
    FormScore
End Sub
```

```
Private Sub mnu_Game_Click()
    ClickNumber = 0
    Form1.Caption = "Memory Game"
End Sub
```

```
Private Sub mnu_Quit_Click()
  End
End Sub
```

טופס מס׳ 2

```
Private Sub mnu_Show_Click()
        txt_Score.Text = GetSetting(appname:="MemoryGame", _
        section:="Scores", _
        Key:=txt_Name.Text, Default:="")
End Sub
```

```
Private Sub cmd_OK_Click()
If chk_Save.Value = Checked Then
        SaveSetting appname:="MemoryGame", _
        section:="Scores", _
        Key:=txt_Name.Text, _
        Setting:=Form1.ClickNumber
End If
        mnu_Show.Enabled = True
        Label1.Caption = "Player's name"
        Label2.Visible = True
        txt_Score.Visible = True
   cmd_OK.Visible = False
   chk_Save.Visible = False
End Sub
```

```
Private Sub txt_Name_Change()
 cmd_OK.Enabled = True
End Sub
```

```
Private Sub mnu_Exit_Click()
 Form2.Hide
End Sub
```

מודול 1

```
Sub FormScore()
  Form2.Show
   With Form2
      .Label1.Caption = "Enter your name"
      .txt_Name.Text = ""
      .txt_Score.Text = ""
      .chk_Save.Visible = True
      .chk_Save.Value = Unchecked
      .cmd_OK.Visible = True
   End With
End Sub
```

9 משחק "קוד סודי - 2"

בשני הפרקים הבאים של הספר נציג את שני הפרויקטים הטובים ביותר שבוצעו על ידי תלמידי בית הספר למחשבים. הפרויקטים נכתבו על ידי הילדים, ללא שום עזרה מצידנו. אנו מביאים אותם כמות שהם. לכן, אם מישהו מקוראי הספר ימצא אי-דיוקים או "באגים", או שאולי למישהו יהיה רעיון כיצד לשפר את התוכניות, נקבל בשמחה את כל ההערות, השיפורים והתוספות.

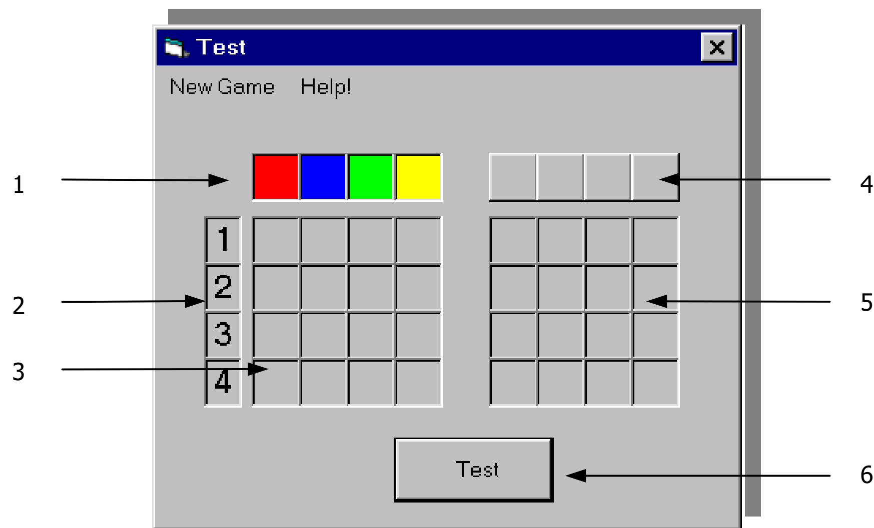
משחק "קוד סודי - 2"

הפרויקט פותח על ידי דניאל פפר בן 14, שמוכר לנו כבר מפרקי הספר הקודמים. את רעיון המשחק "קוד סודי - 2" הוא שאב מהפרויקט שלנו (ראה פרק 6), אך כתב את התוכנית בצורה עצמאית לחלוטין.

נראה כיצד עשה זאת.

מרכיבי ממשק המחשב

נצייר מסגרת - **Frame1** ונמקם עליה את כל המרכיבים, כפי שנראה בתמונה:



1. תיבות **PictureBox** בשמות **picRed**, **picBlue**, **picGreen** ו-**picYellow** שצבעונות באדום, כחול, ירוק וצהוב (מאפיין **BackColor**). המאפיין **DragMode** נקבע במצב **1-Automatic**. לתיבות מוקצים תגים (**Tags**) **1**, **2**, **3** ו-**4**.
2. תוויות רגילות (**Labels**) עם מספרים (**Captions**) מ-**1** עד **4**.
3. ארבעה מערכי תיבות **PictureBox** בשמות:

- **picNum1 (0)... picNum1 (3)** (שורה ראשונה)
- **picNum2 (0)... picNum2 (3)** (שורה שנייה)
- **picNum3 (0)... picNum3 (3)** (שורה שלישית)
- **picNum4 (0)... picNum4 (3)** (שורה רביעית)

המאפיין **DragMode** נקבע במצב **1-Automatic**.

4. מערך מ-4 **CommandButton** בשמות **cmdRealNum(0)...cmdRealNum(3)** ועם מאפיין **Enabled** - **False**.

5. ארבעה מערכי תוויות **Labels** בשמות:

- **lblNum1 (0)... lblNum1 (3)** (שורה ראשונה)
- **lblNum2 (0)... lblNum2 (3)** (שורה שנייה)
- **lblNum3 (0)... lblNum3 (3)** (שורה שלישית)
- **lblNum1 (0)... lblNum 4(3)** (שורה רביעית)

6. לחצן פקודה **CommandButton** בשם **cmdOK**.

מתחת למסגרת **Frame1** נמצאות שתי תוויות עם הכיתוב:



בתמונה מוצגות התוויות בנפרד, בפרויקט הן מוצבות אחת על השנייה, כדי שהכיתוב יהיה במרכז. שמות התוויות: **lblWon (Very good! You win!)** ו-**lblGmOvr (Game Over)**.

7. יש להסיר את כל 4 הלחצנים **cmdRealNum(0) ... cmdRealNum(3)** מהמסגרת-**Frame1** בעזרת פקודת **Cut** ולהעבירו לטופס בעזרת פקודת **Paste**.

התפריט כולל שני חלקים - **New Game** (**mnuNew**) ו- **Help!** (**mnuHelp**)

```
Dim Numbers(3), i, Row As Integer

Private Sub Form_Load()
Row = 1
For i = 0 To 3
    Randomize
    Numbers(i) = Int((Rnd * 4) + 1)
Next
End Sub

Private Sub cmdOK_Click()
Select Case Row
    Case 1
        For i = 0 To 3
            lblNum1(i).BackColor = vbBlack
            If picNum1(i).Tag = Numbers(i) Then _
              lblNum1(i).BackColor = vbWhite
        Next
    Case 2
        For i = 0 To 3
            lblNum2(i).BackColor = vbBlack
            If picNum2(i).Tag = Numbers(i) Then _
              lblNum2(i).BackColor = vbWhite
        Next
    Case 3
        For i = 0 To 3
            lblNum3(i).BackColor = vbBlack
            If picNum3(i).Tag = Numbers(i) Then _
              lblNum3(i).BackColor = vbWhite
        Next
    Case 4
        For i = 0 To 3
            lblNum4(i).BackColor = vbBlack
            If picNum4(i).Tag = Numbers(i) Then _
              lblNum4(i).BackColor = vbWhite
        Next
End Select
```

```
CheckIfWin
Row = Row + 1
If Row = 5 Then
    Frame1.Visible = False
    lblGmOvr.Visible = True
End If
End Sub
```

```
Private Sub mnuNew_Click()
Form_Load
For i = 0 To 3
    picNum1(i).Tag = 0
    picNum1(i).BackColor = vbButtonFace
    picNum2(i).Tag = 0
    picNum2(i).BackColor = vbButtonFace
    picNum3(i).Tag = 0
    picNum3(i).BackColor = vbButtonFace
    picNum4(i).Tag = 0
    picNum4(i).BackColor = vbButtonFace
    lblNum1(i).BackColor = vbButtonFace
    lblNum2(i).BackColor = vbButtonFace
    lblNum3(i).BackColor = vbButtonFace
    lblNum4(i).BackColor = vbButtonFace
    cmdRealNum(i).BackColor = vbButtonFace
Next
lblWon.Visible = False
lblGmOvr.Visible = False
Frame1.Visible = True
End Sub
```

```
Private Sub picNum1_DragDrop(Index As Integer, Source As Control, _
  X As Single, Y As Single)
If Source.Tag <> "" Then
    picNum1(Index).BackColor = Source.BackColor
    picNum1(Index).Tag = Source.Tag
End If
End Sub
```

```
Private Sub picNum2_DragDrop(Index As Integer, Source As Control, _
   X As Single, Y As Single)
If Source.Tag <> "" Then
    picNum2(Index).BackColor = Source.BackColor
    picNum2(Index).Tag = Source.Tag
End If
End Sub
```

```
Private Sub picNum3_DragDrop(Index As Integer, Source As Control, _
  X As Single, Y As Single)
If Source.Tag <> "" Then
    picNum3(Index).BackColor = Source.BackColor
    picNum3(Index).Tag = Source.Tag
End If
End Sub
```

```
Private Sub picNum4_DragDrop(Index As Integer, Source As Control, _
   X As Single, Y As Single)
If Source.Tag <> "" Then
    picNum4(Index).BackColor = Source.BackColor
    picNum4(Index).Tag = Source.Tag
End If
End Sub
```

```
Sub CheckIfWin()
If lblNum1(0).BackColor = vbWhite And _
  lblNum1(1).BackColor = vbWhite And _
  lblNum1(2).BackColor = vbWhite And _
  lblNum1(3).BackColor = vbWhite Or _

  lblNum2(0).BackColor = vbWhite And _
  lblNum2(1).BackColor = vbWhite And _
  lblNum2(2).BackColor = vbWhite And _
  lblNum2(3).BackColor = vbWhite Or _

  lblNum3(0).BackColor = vbWhite And _
  lblNum3(1).BackColor = vbWhite And _
  lblNum3(2).BackColor = vbWhite And _
  lblNum3(3).BackColor = vbWhite Or _

  lblNum4(0).BackColor = vbWhite And _
  lblNum4(1).BackColor = vbWhite And _
  lblNum4(2).BackColor = vbWhite And _
```

```
  lblNum4(3).BackColor = vbWhite Then _
   Frame1.Visible = False
   lblWon.Visible = True
   Row = 1
   For i = 0 To 3
       Select Case Numbers(i)
           Case 1
              cmdRealNum(i).BackColor = vbRed
           Case 2
             cmdRealNum(i).BackColor = vbBlue
         Case 3
            cmdRealNum(i).BackColor = vbGreen
         Case 4
           cmdRealNum(i).BackColor = vbYellow
       End Select
   Next
End If
End Sub
```

```
Private Sub mnuHelp_Click()
         MsgBox "WHITE = Right answer" & Chr(13) & "BLACK = Wrong answer"
End Sub
```

הסברים לפרויקט

1. בפרוצדורת טעינת הטופס (**Form_Load**) מוקצים לארבעת מרכיבי (**3...0**) המערך **Numbers(3)** ערכים אקראיים מ-**1** עד **4**.
2. לאחר מכן, בפרוצדורה **Sub CheckIfWin()** נשתמש בערכים אלה, כדי לקבל צבעים אקראיים של הלחצנים **cmdRealNum**:

```
For i = 0 To 3
        Select Case Numbers(i)
            Case 1
                cmdRealNum(i).BackColor = vbRed
            Case 2
                cmdRealNum(i).BackColor = vbBlue
            Case 3
                cmdRealNum(i).BackColor = vbGreen
            Case 4
                cmdRealNum(i).BackColor = vbYellow
        End Select
Next
```

הביטוי **Select Case** בשפת Visual Basic מקביל לתנאי **If .. Then**. כלומר, ניתן לכתוב גם כך:

```
For i = 0 To 3
  If Numbers(i) = 1 Then _
     cmdRealNum(i).BackColor = vbRed
  If Numbers(i) = 2 Then _
     cmdRealNum(i).BackColor = vbBlue
  If Numbers(i) = 3 Then _
     cmdRealNum(i).BackColor = vbGreen
    If Numbers(i) = 4 Then _
       cmdRealNum(i).BackColor = vbYellow
 Next
```

במקום להשתמש בדרך זו, דניאל בוחן 4 מקרים כאשר מרכיב המערך **Numbers (i)** מקבל ערך אקראי מ-**1** עד **4** (**Case 1 .. Case 4**). תחביר הפרוצדורה הזה מתאים יותר להיגיון הפרוצדורה, מאשר תחביר התנאי **If .. Then**. כך "נשמע" היגיון הפרוצדורה בשפה רגילה:

בשביל משתנה **i** מ-**0** עד **3**

נבחר מקרה, כאשר המרכיב מספר **i** של המערך **Numbers(i)** יקבל ערך מסוים

במקרה, כאשר **Numbers (i) = 1**

צבע הלחצן **cmdRealNum(i)** יהיה אדום

במקרה, כאשר **Numbers (i) = 2**

צבע הלחצן **cmdRealNum(i)** יהיה כחול

במקרה, כאשר **Numbers (i) = 3**

צבע הלחצן **cmdRealNum(i)** יהיה ירוק

במקרה, כאשר **Numbers (i) = 4**

צבע הלחצן **cmdRealNum(i)** יהיה צהוב

נפסיק בחירה

נעבור לערך הבא של המשתנה i.

מעניין, שבמהלך המשחק עצמו הלחצנים **cmdRealNum** כמעט ואינם משתתפים בפועל. רק במקרה ניצחון הם מקבלים צבעים בהתאם לערכים של מרכיבי המערך **Numbers (i)** ומופיעים על המסך יחד עם הכיתוב "**Very good! You win!**". פעולה זו מוגדרת באמצעות התנאי **If .. Then** שבתחילת הפרוצדורה **Sub CheckIfWin()**.

"הקוד הסודי" נרשם במרכיבי המערך **Numbers (i)**. ובכן, מה מתרחש בזמן ניחוש הקוד?

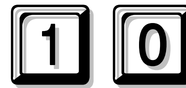
אנו מכירים כבר את האירוע **DragDrop** שמאפשר "למשוך" את מאפייני מרכיב אחד למרכיב אחר. כשאנו "מושכים" בעזרת העכבר את התיבה הצבעונית (**PictureBox**) אל אחד מחלונות המערך **picNum1** (או **picNum2** וכו'), יחד עם הצבע מועבר לחלון המערך גם התג (**Tag**).

```
Private Sub picNum1_DragDrop(Index As Integer, Source As Control, _
X As Single, Y As Single)
If Source.Tag <> "" Then
    picNum1(Index).BackColor = Source.BackColor
    picNum1(Index).Tag = Source.Tag
End If
End Sub
```

לאחר מכן, התג (**Tag**) מושווה לערך אקראי של מרכיב המערך **Numbers (i)**, ותווית המערך **lblNum1(i)** התואמת למרכיב זה, נצבעת בצבע לבן (שאר התוויות נצבעות בצבע שחור):

```
If picNum1(i).Tag = Numbers(i) Then _
   lblNum1(i).BackColor = vbWhite
```

הניצחון נקבע כשכל התוויות (**lblNum1(i** שבאחת מארבע השורות תיצבענה בצבע לבן, כתוצאת מהבדיקה המתבצעת בפרוצדורה **CheckIfWin()**.



משחק "שולה מוקשים"

הפרויקט פותח על ידי אלכס פרדקין, בן 15. עד לפני שנה לא ידע אלכס לתכנת. במשך לימודיו הוא פיתח מספר פרויקטים מעניינים מאוד בשפות Pascal ו- Visual Basic.

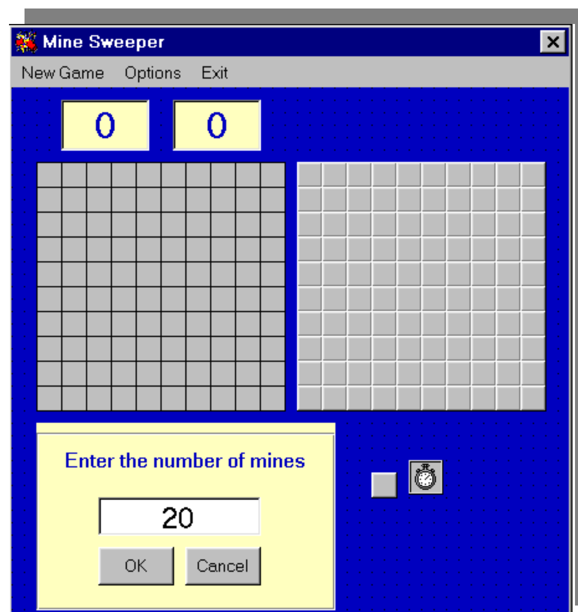
המשחק "שולה מוקשים" (באנגלית "Minesweeper") מוכר כבר מגרסת Windows 3.1. אלכס השתמש ברעיון המשחק וכתב תוכנית משלו. המשחק פחות מורכב מהמשחק האמיתי, אך מעניין למדי והניצחון מחייב מאמצים רציניים.

אלכס כתב תוכנית גדולה ודי מסובכת. כדי להקל קצת, נציג אותה עם מספר קיצורים. למשל, לא הכנסנו לטקסט התוכנית את הפרוצדורות לשמירת תוצאות המשחק. אך בפרקים 7 ו-8 למדתם כיצד עושים זאת.

נראה כיצד עובדת התוכנית שלו.

מרכיבי ממשק המשחק

1. ניצור שני מערכים גדולים: מערך לחצנים (**CommandButton**) ומערך תוויות (**Labels**). כל מערך מורכב מ-160 רכיבים: **cmd_Mine(0)...cmd_Mine(159)** ו- **lbl_Mine(0)... lbl_Mine(159)**.
2. נקבע את מאפייני הלחצנים **cmd_Mine**, גדלים: **252 x 252**, מאפיין **Style** כ-**Graphical**.
3. למאפיין **Appearance** של תוויות **lbl_Mine** נקצה ערך **0-Flat** ונצבע אותן בצבע אפור (**Button Face**).
4. נציב על הטופס את הלחצנים ואת התוויות בצורת רשתות המורכבות ממאה משבצות (10 x 10), החל מאינדקס **30** ועד לאינדקס ***129***. את שאר הלחצנים נעביר לחלק הטופס שיהיה חסוי במהלך המשחק.



5. ניצור מסגרת (**Frame**) בשם **frm_Mines**, נציב עליה: תווית עם הכיתוב "**Enter the number of mines**", תיבת טקסט **txt_Mines** עם ערך **20** במאפיין **Text**, זה מספר המוקשים שיהיה ברירת מחדל, אלא אם השחקן יבחר אחרת. בתחילת המשחק נהפוך את המסגרת לבלתי-גלויה.

6. מתחת למסגרת נוסיף שני לחצנים: **cmd_OK** ו-**cmd_Cancel**.

7. בחלק התחתון של הטופס נציב שתי תוויות-Labels לחישוב המוקשים המסומנים (**lbl_Mark**) וזמן המשחק (**lbl_Time**).

8. נוסיף לטופס טיימר **Timer1** למדידת הזמן. את המאפיין **Enabled** נגדיר כ-**False** ונקבע **Interval - 1000** (שנייה אחת).

9. להסרת הפוקוס, נוסיף בתחתית הטופס לחצן **cmd_Focus**.

10. נבנה את התפריט:

New game		**mnu_New**
Options		**mnu_Options**
	Set number of mines	**itm_Mines**
Exit		**Mnu_Exit**

תוכנית המשחק והערות

```
Dim GameTime, i, j, MinesNumber, Mines, Marker_Counter, Num As Integer
```

בפרוצדורת טעינת הטופס שני המשתנים **MinesNumber** ו-**Mines** מקבלים ערכים השווים למספר המוקשים שהוזן לתיבת הטקסט. אחר, בלולאה **Do Until ... Loop**, מספר המוקשים שהוזן נטען למשבצות אקראיות של תוויות **lbl_Mine**. המשבצות נצבעות בכחול (**vbBlue**) והתוויות סביב המוקשים מסומנות בתג (**Tag**) "**Danger**".

```
Private Sub Form_Load()
MinesNumber = txt_Mines.Text
Mines = txt_Mines.Text
Do Until MinesNumber = 0
   Randomize
   j = Int(Rnd * 100) + 30
    If lbl_Mine(j).Caption = "" Then
      lbl_Mine(j).BackColor = vbBlue
      lbl_Mine(j + 1).Tag = "Danger"
      lbl_Mine(j - 1).Tag = "Danger"
      lbl_Mine(j + 9).Tag = "Danger"
      lbl_Mine(j - 9).Tag = "Danger"
      lbl_Mine(j + 10).Tag = "Danger"
      lbl_Mine(j - 10).Tag = "Danger"
      lbl_Mine(j + 11).Tag = "Danger"
      lbl_Mine(j - 11).Tag = "Danger"
      MinesNumber = MinesNumber - 1
    End If
   Loop
End Sub
```

החלק המסובך של התוכנית מתבצע בפרוצדורת הלחיצה על הלחצן cmd_Mine. ננתח אותו צעד-אחר-צעד.

```
Private Sub cmd_Mine_Click(Index As Integer)
cmd_Focus.SetFocus
Timer1.Enabled = True
Num = 0
```

נגדיר תנאי האומר שאם מתבצעת לחיצה על משבצת שערך המאפיין **Tag** שלה אינו **Danger**, היא תיעלם:

```
If lbl_Mine(Index).Tag <> "Danger" Then
   cmd_Mine(Index).Visible = False
   lbl_Mine(Index).Visible = False
End If
```

אחר מופיעה סדרת תנאים המבצעים השוואה, אם השוויון מתקיים, משתנה Num מגדיל את ערכו. ערך זה יופיע מאוחר יותר כמספר על תווית **lbl_Mine**, המציג את מספר המוקשים שנמצאים ליד הלחצן שנלחץ. לכאורה, נראית התוכנית מסובכת ומבולבלת, אך במבט שני תבינו שאלכס ציין נכון את כל התנאים. המספר **240** שמופיע במספר תנאים, תלוי בגודל התווית **lbl_Mine - 252 x 252 twips** (**twips** - יחידות מדידה של קואורדינטות המרכיב על המסך). כדי להגיע לתווית הנמצאת באלכסון, יש להגדיל או להקטין את הקואורדינטות **Top** ו-**Left** ב- **240 twips** ולא ב-**252** כפי שנובע ממידות התווית, משום שיש להפחית **12** יחידות שהן גודל גבולות (**Borders**) התוויות.

```
If _
        lbl_Mine(Index + 1).Top = lbl_Mine(Index).Top And _
        lbl_Mine(Index + 1).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index - 1).Top = lbl_Mine(Index).Top And _
        lbl_Mine(Index - 1).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index + 10).Left = lbl_Mine(Index).Left _ And lbl_Mine
        (Index + 10).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index - 10).Left = lbl_Mine(Index).Left _ And  lbl_Mine
        (Index - 10).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index - 9).Left = _
        lbl_Mine(Index).Left + 240 And _
        lbl_Mine(Index - 9).Top = _
        lbl_Mine(Index).Top - 240 And _
         lbl_Mine(Index - 9).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index + 9).Left = _
        lbl_Mine(Index).Left - 240 And _
        lbl_Mine(Index + 9).Top = _
        lbl_Mine(Index).Top + 240 And _
        lbl_Mine(Index + 9).BackColor = vbBlue Then _
          Num = Num + 1
If _
        lbl_Mine(Index + 11).Left = _
        lbl_Mine(Index).Left + 240 And _
        lbl_Mine(Index + 11).Top = _
        lbl_Mine(Index).Top + 240 And _
        lbl_Mine(Index + 11).BackColor = vbBlue Then _
```

```
        Num = Num + 1
If _
      lbl_Mine(Index - 11).Left = _
      lbl_Mine(Index).Left - 240 And _
      lbl_Mine(Index - 11).Top = _
      lbl_Mine(Index).Top - 240 And _
      lbl_Mine(Index - 11).BackColor = vbBlue Then _
        Num = Num + 1
```

אם הגענו למשבצת ריקה, הלחצן והתווית נעלמים:

```
If Num = 0 Then
  cmd_Mine(Index).Visible = False
  lbl_Mine(Index).Visible = False
End If
```

אם משבצת ריקה (אין בה מוקש), אך המשתנה **Num** אינו שווה לאפס, ערכו יירשם בתווית **lbl_Mine** והלחצן שמעליה ייעלם. צבע הספרה יתחלף בהתאם למספר המוקשים שסביב המשבצת (משתנה **Num**). את בחירת הצבע נבצע באמצעות הפקודה **Select Case**.

```
If lbl_Mine(Index).Caption = "" Then
  lbl_Mine(Index).Caption = Num
  cmd_Mine(Index).Visible = False
  Select Case Num
    Case 1
      lbl_Mine(Index).ForeColor = vbRed
    Case 2
      lbl_Mine(Index).ForeColor = vbBlue
    Case 3
      lbl_Mine(Index).ForeColor = vbGreen
    Case 4
      lbl_Mine(Index).ForeColor = vbCyan
    Case 5
      lbl_Mine(Index).ForeColor = vbButtonFace
    Case 6
      lbl_Mine(Index).ForeColor = vbWhite
    Case 7
      lbl_Mine(Index).ForeColor = vbMagenta
    Case 8
      lbl_Mine(Index).BackColor = vbBlack
  End Select
End If
```

אם עלינו על מוקש, המשחק נגמר. להלן פרוצדורה פשוטה בשם **Sub HideAll()**, ההופכת את כל הלחצנים לבלתי-נראים ומאפשרת למשתמש גישה לבחירת מספר המוקשים במשחק החדש.

```
If lbl_Mine(Index).BackColor = vbBlue Then
        Timer1.Enabled = False
        lbl_mine(Index).Caption = ""
        lbl_Mine(Index).Visible = True
        lbl_Mine(Index).BackColor = vbRed
        HideAll
        Form1.Caption = "Game over"
End If
End Sub
```

```
Sub HideAll()
        itm_Mines.Enabled = True
        For i = 30 To 129
          cmd_Mine(i).Visible = False
        Next
End Sub
```

פרוצדורה מעניינת זו היא בעלת מספר תנאי **If** אחד בתוך השני, ומאפשרת לנו להציב על הלחצן את דמות הדגל באמצעות לחיצה על הלחצן הימני של העכבר. אם מתחת ללחצן מסתתר מוקש, ספירת המוקשים המנוטרלים בחלון **lbl_Mark** תגדל באחד.

```
Private Sub cmd_mine_MouseDown(Index As Integer, _
   Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then
        If cmd_Mine(Index).Picture = LoadPicture() Then
          cmd_Mine(Index).Picture = LoadPicture("flag.bmp")
          If lbl_Mine(Index).BackColor = vbBlue Then
              lbl_Mine(Index).Caption = ""
              Marker_Counter = Marker_Counter + 1
              lbl_Mark.Caption = Marker_Counter
          End If
        End If
End If
End Sub
```

הלחצן **OK** מחזיר את מרכיבי המשחק למצבם ההתחלתי. חלק מפעולות אלו מתבצעות גם בפרוצדורה לתפריט **New game**. שימו לב, מהפרוצדורה **Private Sub cmd_OK_Click()** ניתן לזמן את הפרוצדורה **Form_Load**. שיטה זו מאפשרת לא לכתוב מחדש את הפקודות שנכתבו כבר בפרוצדורה **Private Sub Form_Load()**.

```
Private Sub cmd_OK_Click()
    MinesNumber = txt_Mines.Text
    Mines = txt_Mines.Text
    frm_Mines.Visible = False
    Marker_Counter = 0
    Form1.Caption = "Mine Sweeper"
    For i = 30 To 129
        cmd_Mine(i).Enabled = True
        cmd_Mine(i).Caption = ""
        cmd_Mine(i).Visible = True
        lbl_Mine(i). BackColor = vbMenuBar
        lbl_Mine(i).Visible = True
    Next
    Form_Load
End Sub
```

```
Private Sub mnu_New_Click()
    frm_mines.Visible = True
    itm_Mines.Enabled = True
    For i = 30 To 129
      With cmd_mine(i)
       .Enabled = True
       .Visible = True
       .Picture = LoadPicture()
      End With
      With lbl_mine(i)
       .FontName = Screen.Fonts(5)
       .BackColor = vbBlue
      End With
  Next
  Timer1.Enabled = False
  lbl_time.Caption = "0"
  lbl_mark.Caption = "0"
  GameTime = 0
End Sub
```

הפעולות שמתבצעות על ידי לחצן **Cancel**, פריט התפריט **Set number of mines** והטיימר אינן זקוקות להערות.

```
Private Sub cmd_Cancel_Click()
    frm_Mines.Visible = False
End Sub
```

```
Private Sub itm_Mines_Click()
  frm_Mines.Visible = True
  itm_Mines.Enabled = False
End Sub
Private Sub Timer1_Timer()
    GameTime = GameTime + 1
    lbl_Time.Caption = GameTime
End Sub
```

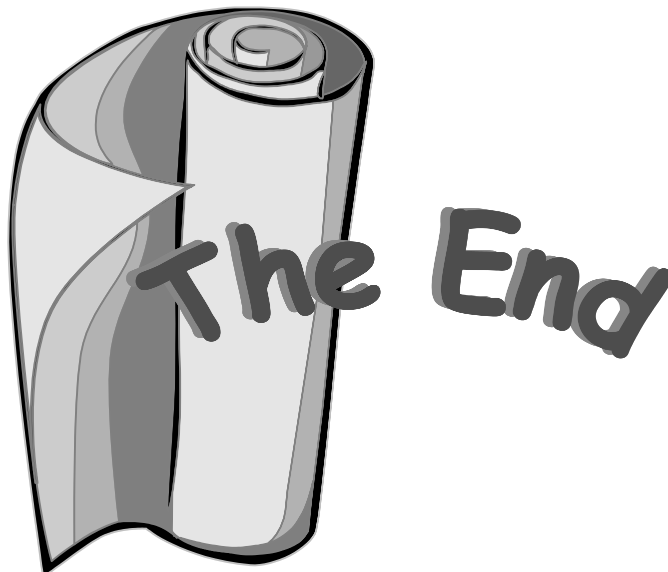
כאשר מספר המוקשים המנוטרלים שווה למספר שהוזן בתחילת המשחק, הפרוצדורה **Private Sub lbl_mark_Change()** תודיע לנו על כך:

```
Private Sub lbl_mark_Change()
If lbl_Mark.Caption = Mines Then
    HideAll
    Form1.Caption = "You win!"
End If
End Sub
```

לבסוף, תפריט Exit עם פקודה פשוטה ביותר - **End**.

```
Private Sub mnu_exit_Click()
    End
End Sub
```

הספר הגיע לסיומו, ואם נהניתם לקרוא אותו - הרי שהמחברים הצליחו במשימתם.



נספח: התקליטור המצורף

בתקליטור המצורף לספר זה תוכל למצוא מספר דברים:

- **קטלוג CD** - קטלוג ספרי המחשבים הממוחשב האינטראקטיבי של **הוצאת הוד-עמי** (נדרשת התקנת תוכנה; כולל חיפוש, הדפסה, דפדוף).
- **קטלוג HTML** - קטלוג ספרי המחשבים האינטראקטיבי של **הוצאת הוד-עמי** (**לא** נדרשת התקנת תוכנה; מומלץ לצפייה עם IE מגירסה 4 ומעלה).
- **ספרים לדוגמה** בהם ניתן לעיין וגם להדפיס.
- תוכנת **מבחן אישי** - לבדיקת ידע בתוכנות Office ובה מאגר של מעל 400 שאלות.
- גיליון מלא של ירחון **חושבים חלונות** ובו מאמרים ב- Windows 95, Word 97 ו-Excel 97.
- מספר תוכנות עזר שימושיות.
- קבצי תרגול.

אם מנהל התקן כונן התקליטורים המותקן הוא 16 סיביות (וזה יכול להיות גם אם הכונן חדש) - ייתכן שתראה רק את 8 התווים הראשונים של שם הקובץ (במקרה שהמקור ארוך יותר), או ייתכן שתראה שהתיקיות ריקות.

הטוב ביותר הוא להתקין מנהל התקן 32 סיביות, או לקנות כונן תקליטורים חדש ולוודא שמצורף אליו מנהל התקן 32 סיביות.

קטלוג CD - הקטלוג הצבעוני האינטראקטיבי של הוד-עמי

הוצאת הוד-עמי גאה לבשר על הקטלוג הצבעוני האינטראקטיבי היחידי בארץ בתוכנה!

הקטלוג **הצבעוני** האינטראקטיבי של הוד-עמי עושה שימוש בטכנולוגיית IBL (Internet/Intranet Based Learning), שהינה המילה האחרונה בכל הקשור לשימושים ולאפשרויות חדשות הגלומות בשימוש בטכנולוגיות אינטרנט.

בעזרת הקטלוג הצבעוני האינטראקטיבי של הוד-עמי תוכל:

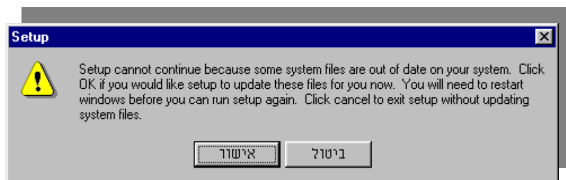
- לעיין במידע על ספרי ההוצאה מתי שתרצה (לאחר התקנת התוכנה, בחר **התחל**, **תוכניות**, **קטלוג הוד-עמי**).
- לעבור במהירות ובקלות בין הקטלוג הצבעוני האינטראקטיבי של הוד-עמי והיישום בו אתה עובד.
- לדפדף בקטלוג **הצבעוני** האינטראקטיבי ספר אחר ספר, או לפי נושאים וקבוצות.
- להדפיס את המחירון המלא ומידע על כל ספר.
- לגשת במהירות, בגישה אינטואיטיבית, תוך התמקדות מהירה בספר המבוקש.
- לעיין בקטלוג **הצבעוני** האינטראקטיבי בקצב אישי שלך.
- לנווט את דרכך בקטלוג **הצבעוני** האינטראקטיבי ולחזור ולהתרענן בכל נושא בכל רגע.
- להוריד עדכונים אודות ספרים חדשים מהאינטרנט בכתובת:

http://www.hod-ami.co.il/

הקטלוג מתאים לעבודה ב-Windows 95/98 עם Internet Explorer 3.02a בגירסה העברית, אך עם זאת, ניתן לצפייה גם עם דפדפן MS-IE4 ומעלה.

הקטלוג מתאים לעבודה ב-Windows 98 אם כי ההתקנה דורשת התערבות מצידך כדי שתושלם. לפרטים קרא בקובץ README הנמצא בתיקיה X:\Catalog.

1. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד את הפקודה **X:\Catalog\Hod-ami** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.



3. עקוב אחר ההוראות המופיעות על המסך ופעל לפיהן. כאשר מופיעה תיבת ההודעה **Setup** המודיעה על התקנת רכיבים חדשים - לחץ **אישור**. פעולה זו מצריכה אתחול מחדש של המחשב.

3. לאחר שהמחשב עולה מחדש הפעל את תוכנית ההתקנה מחדש:

- לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
- בתיבת הטקסט הקלד את הפקודה **X:\Catalog\Hod-ami** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.

5. לחץ על לחצן **Setup**. מופיע חלון עם רקע כחול ובו הכותרת: "**התקנת קטלוג הוד-עמי**". לחץ **המשך**.



6. בחלון **התקנת קטלוג הוד-עמי** לחץ על תמונת המחשב. אם מופיעה תיבת דו-שיח בה תתבקש להגדיר היכן להתקין את Internet Explorer, השתמש בברירת המחדל. לא להבהל, זו אינה התקנה "אמיתית" של Internet Explorer, אלא רק של מספר רכיבים הדרושים להפעלת הקטלוג (שים לב, שאין צורך בהתקנה מלאה של Internet Explorer). לחץ **OK**.

6. בסיום ההתקנה כולה תופיע תיבת דו-שיח. לחץ על **אישור**. יופיע חלון **Setup**, אשר את אתחול המחשב פעם נוספת על ידי בחירה בלחצן **כן**.

7. בחר בתפריט **התחל** באפשרות **הפעלה**.

8. בתיבת הטקסט הקלד **X:\Catalog\Update9809.exe** (החלף את האות X באות הכונן המתאים) ולחץ על **אישור**. זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.

9. קרא את ההודעה ולחץ **אישור**.

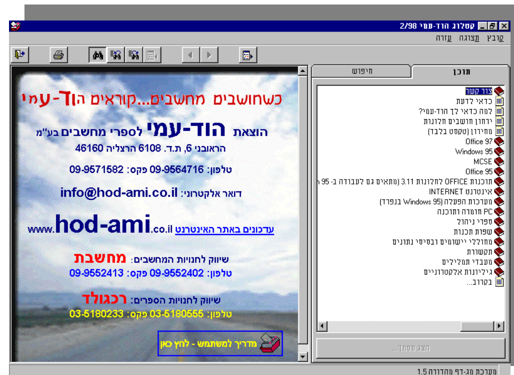
10. בחלון WinZip Self-Extractor בחר בלחצן **Unzip** והמתן.

11. קרא את ההודעה ולחץ **אישור**.

12. בחלון WinZip Self-Extractor בחר בלחצן **Close**.

13. חזור על סעיפים 8-14 עבור קובץ **X:\Catalog\Update????.exe**. (ארבעת סימני השאלה הם עבור 9903, או מספר גדול יותר כמו 9904, 9905 וכדומה).

14. לאחר שהמחשב עולה מחדש, לחץ על לחצן **התחל**, **תוכניות** ובחר באפשרות **קטלוג הוד-עמי**.



בכל חודש הקטלוג מתעדכן. ניתן להוריד קובץ עדכון מאתר ההוצאה בכתובת:

www.**hod-ami**.co.il

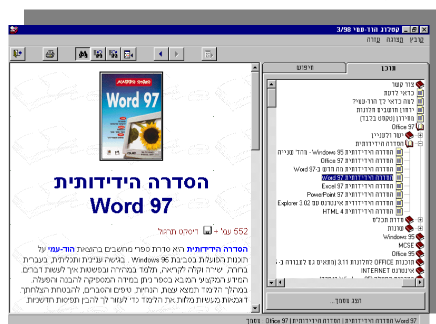
הפעלת התוכנה/הקטלוג

להפעלת הקטלוג הצבעוני של הוצאת **הוד-עמי**, לחץ על **התחל**, עבור ל**תוכניות** ובתתחיתית התפריט תראה את **קטלוג הוד-עמי**. בחר בו.

התקנת הקטלוג ב-Windows 95 עשויה (לא תמיד) לשבש את העבודה עם הכתבן. במקרה זה יש להעתיק את קובץ Riched32.dll לתיקיית המערכת של Windows 95. בדרך כלל, זו תיקיה C:\Windows\System..

מבנה המסך הראשי

בהפעלת התוכנה יופיע מסך ובו שלושה אזורים עיקריים:



- מימין יופיע **סייר המסמכים** (דומה ל**סייר Windows**). למעשה, זהו עץ המציג את פרטי הקטלוג השונים. אזור זה ישמש בהמשך גם לתהליך החיפוש.
- בצד שמאל יופיע דף הפתיחה. זהו **אזור התוכן**.
- בחלקו העליון יופיעו שורת התפריטים וסרגל הכלים.

כך נראה מסך של קטלוג **הוד-עמי** הנמצא בתקליטור המצורף לספר.

סיור מודרך בצעדים קלים

1. לחץ לחיצה כפולה על הסמל שלידו רשום **Office 97**.
2. לחץ לחיצה כפולה על הסמל שלידו רשום **הסדרה הידידותית/קוראים יודעים**.
3. תיפתח רשימת מסמכים הקשורים לנושא.
4. לחץ לחיצה כפולה על הסמל שלידו רשום **קוראים יודעים Word 97**.
5. בצד שמאל יוצג מסמך שכותרתו **קוראים יודעים Word 97**.
6. לחץ על הסמל שבסרגל הכלים שבראש החלון, כדי לעבור לדף הבא.
7. יופיע דף שכותרתו **הסדרה הידידותית הכרת המחשב האישי**.
8. גרור מטה את פס הגלילה של החלון השמאלי, כדי לקרוא את המשך התיאור.
9. אם ברשותך מדפסת, תוכל להדפיס את הדף על ידי לחיצה על הסמל .

שינוי מבנה המסך

ניתן לשנות את תצוגת המסמך על ידי שני לחצנים :


בהצבת הסמן בין אזור סייר המסמכים לאזור המסמך יוצג חץ דו-ראשי. בגרירה ימינה ושמאלה תוכל לשנות את חלוקת המסך בין שני האזורים.


הצג/הסתר סייר מסמכים. בלחיצה על לחצן זה, אזור המסמך "ישתלט" על כל המסך. לחיצה נוספת על לחצן זה תחזיר את המצב לקדמותו.

סרגל הכלים


יציאה מהתוכנה.


הדפס מסמך. המסמך המוצג על המסך יישלח להדפסה.


הצג/הסתר סייר מסמכים. בלחיצה על לחצן זה, אזור המסמך "ישתלט" על כל המסך. לחיצה נוספת על לחצן זה תחזיר את המצב לקדמותו.


הקודם/הבא בסייר המסמכים. בהתאם למיקום הסמן יוצג המסמך הקודם/הבא בעץ המסמכים (ראה פירוט בהמשך).


הצג מיקום המסמך בתוכן. הסמן יתמקם בסייר המסמכים על השורה המתאימה למסמך הנוכחי. שים לב, ייתכן שהסמן בעץ המסמכים ניצב על מסמך שאינו המסמך המופיע באזור התוכן באותו שלב (כדי להתאימם נשתמש בכלי הבא).

מסמך קודם/הבא. דפדוף קדימה ואחורה במסמכים שכבר עברנו עליהם בהפעלה נוכחית (ראה פירוט בהמשך).

הצג תת-עץ במסמך. כשהסמן ניצב על נושא ובו מסמכים ו/או תת-נושאים נוספים, שימוש בלחצן זה באזור סייר המסמכים יפרוס את תת-הנושאים והמסמכים ובצד השני יוצג מסמך אחד ארוך המכיל את כל המסמכים.

דפדוף בעץ המסמכים

ניתן לבצע את פעולת הדפדוף בשלוש דרכים שונות:

- מעבר על עץ המסמכים.
- פתיחת נושא.
- סגירת פרק.

מעבר על עץ המסמכים

- נושא כללי.
- נושא או תת-נושא פתוח.
- ספר.

פתיחת נושא

יש שלוש אפשרויות לפתיחת נושא:

- לחיצה כפולה על שם הנושא.
- לחיצה על סימן "+" המופיע משמאל לסמל הנושא.
- על ידי מקש החיצים: →.

סגירת פרק

יש שלוש אפשרויות לסגירת פרק:

- לחיצה כפולה על שם הנושא.
- לחיצה על סימן "-" המופיע משמאל לסמל הנושא.
- על ידי מקש החיצים: ←.

מעבר בין ענפי עץ המסמכים בעזרת לוח המקשים

- לחיצה על אות כלשהי תעביר את הסמן לשורה הבאה בעץ, המתחילה באותה אות (אם אין שורה כזו, לא יקרה דבר).
- לחיצה על ↓ תעביר את הסמן לשורה הבאה ולחיצה על ↑ תעביר את הסמן לשורה הקודמת בעץ.

הצגת מסמך

יש שתי אפשרויות להצגת מסמך:

- כאשר הסמן ניצב על שם המסמך - לחיצה על לחצן הצג מסמך... .
- לחיצה כפולה על שם המסמך בעץ.

שימוש בלחצנים בסרגל הכלים

דפדוף בעזרת לחצנים אלה דומה לדפדוף בספר, דף אחר דף, קדימה או אחורה. לחצנים אלה משמשים למעבר בין המסמכים, קדימה ואחורה, על פי סדר הופעתם בעץ התפריטים (נקודת ההתחלה היא מיקום הסמן בעץ). אם מגיעים לפרק או לתת-נושא סגור - הוא ייפתח אוטומטית ויוצג המסמך הראשון.

כל לחיצה תפתח את המסמך המתאים (בחלק השמאלי) ותעביר את הסמן לשורה המתאימה בעץ (בחלק הימני).

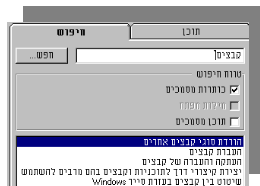
שימוש בלחצנים בסרגל הכלים

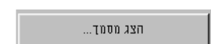
דפדוף בין מסמכים שהופעלו בהפעלה הנוכחית. המסמך הנוכחי הינו המסמך האחרון ממנו אפשר לעבור אחורה עד למסמך הראשון של ההפעלה (מסך הפתיחה). בדרך בין המסמך הראשון למסמך הנוכחי תוכל לדפדף קדימה ואחורה.

הורדת עדכונים

הקטלוג הצבעוני מתעדכן מדי חודש. את העדכון ניתן להוריד מהאינטרנט (חינם!) מאתר ההוצאה. בפתיחת הקטלוג בצד שמאל כתוב **עדכונים באתר האינטרנט**. אם אתה מחובר לאינטרנט, הצב את הסמן על כתובת זו ולחץ. להרחבת התצוגה לחץ על **משקפת**. באתר **הוד-עמי** בחר בקישור **קטלוג והורדת עדכון חודש**. הורד את העדכון, וזכור היכן אתה שומר את הקובץ. צא מהקטלוג ונתק את החיבור לאינטרנט. הפעל את **סייר Windows** ואתר את הקובץ. לחץ עליו לחיצה כפולה. קרא את ההודעה ולחץ **אישור**. לחץ **UnZip**, קרא את ההודעה ולחץ **אישור**. לחץ **Close** והפעל מחדש את הקטלוג. זהו, הקטלוג מעודכן!!!

חיפוש



1. עבור לכרטיסיה **חיפוש** בחלק הימני של החלון.
2. הקלד רצף תווים. זו יכולה להיות מילה, או חלק ממילה.
3. קבע את טווח החיפוש - האם לחפש בכותרות המסמכים ו/או בתוכן המסמכים.
4. הקש **Enter** או לחץ על . אם נמצאו מסמכים, תופענה כותרותיהם.

הצגת המסמך

יש שתי אפשרויות להצגת מסמך:

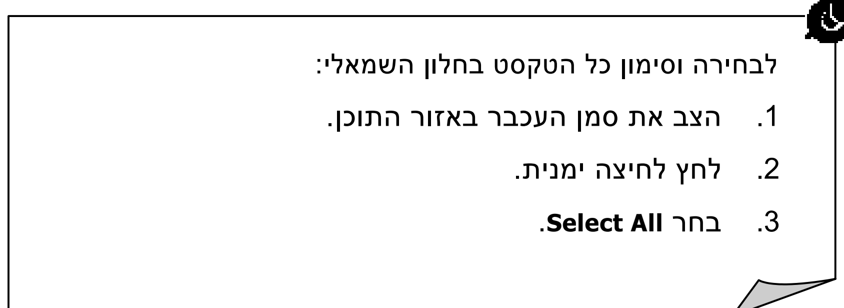
- כאשר הסמן ניצב על שם המסמך - לחיצה על לחצן .
- לחיצה כפולה על שם המסמך.

שים לב! במסמך שמוצג מחלון החיפוש, הטקסט שחיפשת יודגש בצבע אדום.

העתקת טקסט אל מסמכים/תוכניות אחרות

אפשר לסמן קטע ממסמך, להעתיקו ולהדביקו במקום אחר (למשל, במסמך Word).

1. סמן את הטקסט שברצונך להעתיק (הסימון מתבצע על ידי לחיצה וגרירת העכבר על האזור שברצונך להעתיק). הטקסט שסומן ייצבע בצבע כחול.

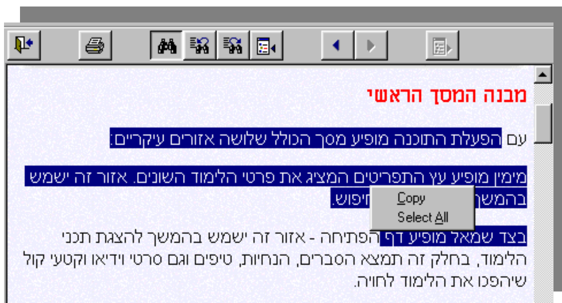


לבחירה וסימון כל הטקסט בחלון השמאלי:

1. הצב את סמן העכבר באזור התוכן.
2. לחץ לחיצה ימנית.
3. בחר **Select All**.

2. ביצוע העתקה:

- הצב את סמן העכבר על הקטע המסומן.
- לחץ לחיצה ימנית ובחר **Copy**, או לחץ **Ctrl+C**.
- עבור לתוכנה אחרת, כמו Word, פנקס רשימות, כתבן וכדומה.



הצב את נקודת הכניסה במקום שבו תרצה להוסיף את הטקסט ובצע אחת מהפעולות הבאות: בחר בתפריט **עריכה**, **הדבק**, או לחץ על לחצן **הדבק** , או הקש **Ctrl+V**, או לחץ לחיצה ימנית ובתפריט המקוצר בחר **הדבק**.

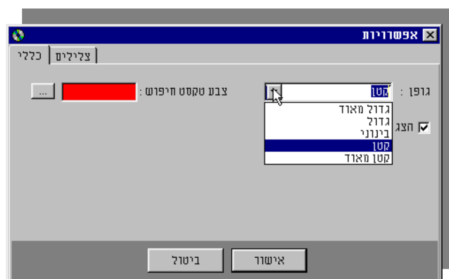
העתקת תמונה לתוכנות אחרות

אפשר להעתיק תמונה ממסמך שמופיע על המסך ולהדביקה במקום אחר (למשל במסמך Word).

1. הצב את הסמן על התמונה שברצונך להעתיק.
2. לחץ לחיצה ימנית ובתפריט המקוצר בחר **Copy**.
3. עבור לתוכנה האחרת: תוכנה גרפית, Word וכדומה.
4. לביצוע ההדבקה, בחר בתפריט **עריכה**, **הדבק**;
   או לחץ על לחצן **הדבק** ;
   או הקש **Ctrl+V**;
   או לחץ לחיצה ימנית ובתפריט המקוצר בחר **הדבק**.

אפשרויות תצוגה

בשורת התפריט בחר **תצוגה**, **אפשרויות**. בחלון זה ניתן לשנות את המאפיינים האלה:



- גודל הגופן המוצג במסמכים השונים.
- צבע טקסט החיפוש (ברירת המחדל - אדום).
- האפשרות להציג שורת מצב, או לא להציגה.
- הצלילים שמשמיעה התוכנה כאשר מתחלפים המסמכים.

בתיבת הדו-שיח **אפשרויות**, תוכל לקבוע גם את צבע טקסט החיפוש.

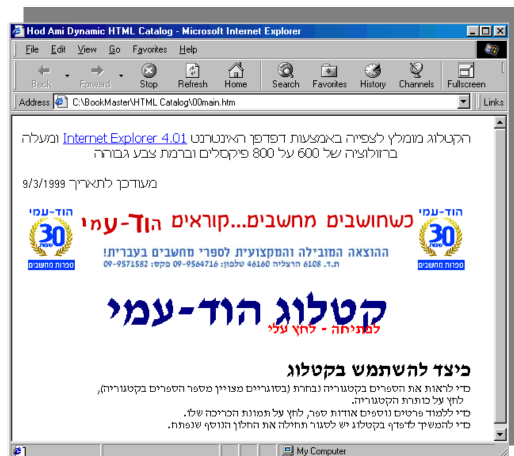
קטלוג HTML - קטלוג צבעוני

הוצאת הוד-עמי גאה לבשר על קטלוג HTML **צבעוני** העושה שימוש בטכנולוגיית DHTML (Dynamic HTML) שהינה המילה האחרונה בעיצוב דפי Web באינטרנט.

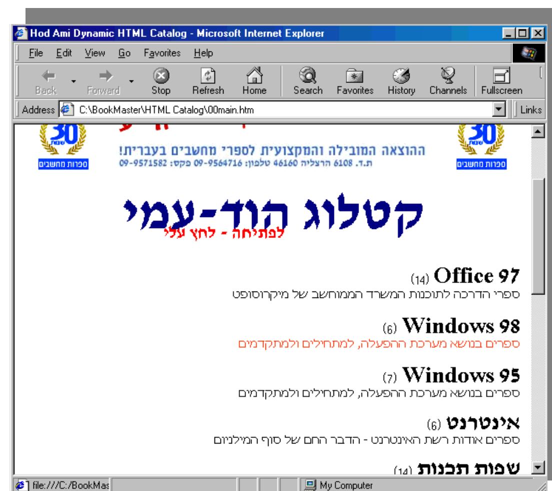
בעזרת קטלוג HTML **צבעוני** תוכל:

- לעיין במידע על ספרי ההוצאה מתי שתרצה (לחיצה כפולה... וזהו!).
- לעבור במהירות ובקלות בין הקטלוג **הצבעוני** והיישום בו אתה עובד.
- לעיין במידע על כל ספר וספר.
- לגשת במהירות, בגישה אינטואיטיבית, תוך התמקדות מהירה בספר המבוקש.
- לעיין בקטלוג **הצבעוני** בקצב אישי שלך.
- לנווט את דרכך בקטלוג **הצבעוני** ולחזור ולהתרענן בכל נושא בכל רגע.

הקטלוג מומלץ לצפייה בעזרת Internet Explorer מגירסה 4 ומעלה.



1. פתח את סייר Windows.
2. עבור לתיקיה HTML Catalog אשר בתקליטור המצורף.
3. אתר את הקובץ main.htm00 והפעל אותו (בדרך כלל על ידי לחיצה כפולה).
4. קרא את ההוראות כיצד להשתמש בקטלוג שבמסך פתיחה זה.
5. לחץ על הכתובת **קטלוג הוד-עמי** לפתיחה - לחץ עלי.
6. יוצגו הנושאים השונים: Office 97, Windows 98 וכדומה.



7. הצב את הסמן על כותרת הנושא, למשל, **Office 97** ולחץ בעכבר.
8. יוצגו הספרים השייכים לאותו נושא.



9. לחץ בעכבר על הספר לגביו אתה מעוניין במידע (שים לב, הסמן **לא** ישתנה).
10. ייפתח חלון בו תוכל לקרוא מידע אודות הספר (מה הוא מכיל, למי הוא מיועד, מה נמצא בתקליטור המצורף, מי המחבר, כמה עמודים בספר וכדומה).

כדי להמשיך לעיין בקטלוג, עליך **לסגור** את החלון המכיל את פרטי הספר.

המחירון המעודכן של ספרי ההוצאה נמצא באתר האינטרנט:

www.hod-ami.co.il

התקנת קטלוג HTML

מי שרוצה להפעיל את קטלוג HTML מתוך התקליטור, צריך לנהוג לפי ההוראות הרשומות לעיל. הפעלה מתוך התקליטור מחייבת את המצאות התקליטור בכונן. בהתקנה פשוטה ניתן להעתיק את הקטלוג לדיסק הקשיח. להלן ההוראות להתקנת קטלוג HTML:

1. לחץ על **התחל** ובחר **הפעלה**.
2. בעזרת לחצן עיון סמן את הקובץ **HodAmiHTMLcatalog.EXE** אשר בתיקיה **X:\HTML Catalog**.
3. לחץ **פתח**.
4. לחץ **אישור** פעמיים.
5. לחץ על לחצן **Unzip**.
6. לחץ אישור.
7. על שולחן העבודה מופיע סמל עם כיתוב קטלוג ספרי מחשבים בהוצאת הוד-עמי.
8. הפעל אותו.



התקנה - Acrobat Reader

תוכנה זו יש להתקין, כדי לקרוא את דוגמת הירחון "**חושבים חלונות**" ואת **הספרים לדוגמה** וגם כדי להפעיל את תוכנות הבחינה לדוגמה המצורפות לתקליטור זה. התוכנה פועלת במערכות הפעלה **Windows 95/98 בלבד!**

1. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד את הפקודה **X:\Software\Adobe Acrobat\Ar32e301** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.
3. בחלון הבא בחר בלחצן **כן** להמשך ההתקנה.

4. אשף ההתקנה מתקין את הרכיבים הנדרשים. עליך ללחוץ על **Next**, **Yes** ו-**Next** פעם נוספת, כדי לסיים את ההתקנה.

5. בסיום ההתקנה לחץ על **Finish**.

6. קרא את הודעת סיום ההתקנה שמופיעה ולחץ **אישור**.

ספרים לדוגמה

בתיקיה **Books DEMO** תמצא מספר דוגמאות מספרי ההוצאה. כדי לראות וגם להדפיס את הקבצים (פורמט pdf), עליך להתקין את Acrobat Reader :

1. פתח את תפריט **התחל**, **תוכניות**, **Adobe Acrobat**, **Acrobat Reader 3.01**.

2. פתח את תפריט **File** ובחר באפשרות **Open**.

3. בתיבה **חפש ב** : בחר בכונן התקליטורים שלך ועבור לתיקיה **Books DEMO**.

4. בחר בקובץ ולחץ על לחצן **Open**.

חושבים חלונות - הפעלה

1. פתח את תפריט **התחל**, **תוכניות**, **Adobe Acrobat**, **Acrobat Reader 3.01**.

2. פתח את תפריט **File** ובחר באפשרות **Open**.

3. בתיבה **חפש ב** : בחר בכונן התקליטורים שלך ועבור לתיקיה **ThinkWin**.

4. בחר בקובץ **ThinkWin - Vol 5.pdf** המופיע בתיקיה, כדי לקרוא גיליון 5 במלואו.

5. בחר בקובץ **ThinkWin - Brief.pdf** המופיע בתיקיה זו, כדי לקרוא את תקצירי הגליונות הקודמים.

ומה הלאה...

הזמנת מנוי - חושבים חלונות

או פקס
09-9571582

שלח בדואר לפקודת הוד-עמי ת.ד. 6108 הרצליה 46160

סמן/י X לפי בחירתך:

אני מעוניין/ת לעשות מנוי על הירחון **חושבים חלונות**.

☐ 12 גיליונות במחיר 99₪ כולל מע"מ

שם משפחה ______________ שם פרטי ______________

כתובת מלאה ______________________________

ישוב ______________________

מיקוד ____________

טלפון ______________ פקס ______________

e-mail: ______________________

ויזה - 16 ספרות   דיינרס - 14 ספרות

ישראכרט - 8 ספרות   מאסטרקארד - 16 ספרות

מספר כרטיס אשראי

בתוקף עד ___ / ___   תעודת זיהוי

כתובת למשלוח התשבוניות (במידה והיא שונה מהכתובת שלעיל):

____

____

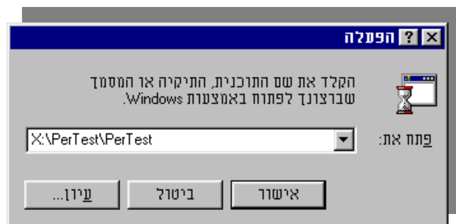
אישור: תאריך   חתימת המזמין

חובה למלא פרטים

מבחן אישי - התקנה

מבחן אישי מאפשר לך לבחון את ידיעותיך בתוכנות Office השונות. בתקליטור המצורף לספר זה תמצא מספר מבחנים לתוכנת Word בגרסאות 7 ו-97.

1. בחר בתפריט **התחל** באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד **X:\PerTest\Pertest.exe** (החלף את האות X באות הכונן המתאימה) ולחץ על **אישור**.



זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.

3. קרא את ההודעה ולחץ **אישור**.
4. בחלון WinZip Self-Extractor בחר בלחצן **Unzip** והמתן.



5. לחץ **אישור**. תהליך ההתקנה מתחיל. המתן.
6. מופיעה הודעה שההתקנה הסתיימה בהצלחה. לחץ **אישור**.

התקנת ערכת מבחנים מלאה ל-Word 7/97

1. בחר בתפריט **התחל** באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד **X:\PerTest\PTWord.exe** (החלף את האות X באות הכונן מתאימה) ולחץ על **אישור**. זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.
3. קרא את ההודעה ולחץ **אישור**.
4. בחלון WinZip Self-Extractor בחר בלחצן Unzip והמתן.
5. קרא את ההודעה ולחץ **אישור**.
6. בחלון WinZip Self-Extractor בחר בלחצן Close.

הפעלה - מבחן אישי הוד-עמי

לחץ על לחצן **התחל**, **תוכניות**, **מבחן אישי - הוד עמי**.

מה עוד בתקליטור?

הוצאת **הוד-עמי** מפיצה תוכנות אלו כבונוס ללקוחות ההוצאה ואינה מתיימרת לגבות תשלום עבור התוכניות המצורפות ו/או לתמוך בהם.



אזהרה!

השימוש בתקליטור זה הוא על אחריותו הבלעדית של המשתמש. המוצרים המותקנים בתקליטור זה מסופקים באחריות החברות המייצרות אותם. הוצאת **הוד-עמי** אינה אחראית, בכל צורה שהיא, לאופן ולטיב התוכנות המותקנות.

בכל שאלה לגבי תוכנה הנמצאת בתקליטור, יש לפנות למפתחי התוכנה (כל תוכנה בנפרד), כפי שמצוין בקבצי העזרה של התוכנה המדוברת.

הקבצים הם גרסאות **שיתופיות** (ShareWare) ו**חופשיות** (FreeWare).

גירסה **שיתופית** (ShareWare) מאפשרת לך, המשתמש, לבדוק את יעילות התוכנה ואת תאימותה לעבודה אותה אתה מבצע. אם נמצאה התוכנה מתאימה לצרכיך, עליך לשלם למפתחיה תשלום סמלי (לפי הרשום בקבצי העזרה של כל תוכנה ותוכנה בנפרד), כדי לקבל רישיון מלא לשימוש בה. רכישת רישיון לשימוש בתוכנה יפתח בפניך מיגוון אפשרויות שייתכן ולא עמדו לרשותך בהפעלת הגירסה השיתופית.

FontsPekan

קובץ זה יתקין במחשב שני גופנים בעברית לשימושך. בסיום ההתקנה יש לבצע את הפעולות הבאות:

1. לחץ על **התחל**, הצבע על **הגדרות** ובחר ב**לוח הבקרה**.
2. לחץ לחיצה כפולה על הסמל **גופנים**.
3. פתח את תפריט **קובץ** ובחר באפשרות **התקנת גופן חדש**.
4. עבור לתיקיה **C:\FontsPekan**.
5. לחץ על לחצן **בחר הכל** (סך הכל יש בתיקיה שני גופנים).
6. ודא שתיבת הסימון **העתק גופנים לתיקיית הגופנים** מסומנת. לחץ **אישור**.
7. סגור את חלון התיקיה **Fonts**.
8. סגור את חלון **לוח הבקרה**.

כעת, מוכנים הגופנים לשימוש בכל התוכנות המותקנות במחשב שלך: Word, Excel, PowerPoint וגם בתוכנות גרפיות, כגון Paint Shop Pro ו-PhotoShop.

הגופנים נקראים Tml-JUMP ו-Tml-step ויופיעו בתחתית רשימת שמות הגופנים (בדרך כלל). הרי דוגמה שלהם:

Tml-step

אבגדהוזחטיכךלמםנןסעפףצץקרשת1234567890

Tml-JUMP

אבגדהוזחטיכךלמםנןסעפףצץקרשת1234567890

Terra - מימד חדש בתצוגה - ים המלח ממעוף הציפור

Terra היא משפחת מוצרים של חברת **סקייליין מערכות תוכנה**, הבנויה סביב מנוע גרפי ייחודי ורב עוצמה, המאפשר תצוגה בזמן אמת של פני שטח בתלת מימד, ללא הגבלת פרטים וגודל, על גבי מחשבים ביתיים. מקור תמונת השטח הוא בתצלומי לווין ובתצלומי אוויר.

דרישות מערכת מינימליות:

- מעבד מסוג Pentium בטכנולוגיית MMX.
- זיכרון פנימי 32MB RAM.
- זיכרון כרטיס מסך 2MB.
- כונן תקליטורים במהירות 6x.
- מערכת הפעלה Windows 95/98.
- התקליטור חייב להיות בכונן במהלך ההפעלה.

מנוע Terra לוקח את התצוגה בזמן אמת לרמות חדשות שלא נודעו בעולם מחשבי ה-PC. הוא מאפשר אין סוף מצבי תצוגה: החל ממבט קרוב לפרטי פרטים ועד תצוגה רחוקה עד האופק. מנוע Terra עושה שימוש בטכנולוגיית MMX החדשנית ומאפשר איכויות שבעבר היו נחלתם של מחשבים יקרים בלבד.

לפרטים נוספים על התוכנה ומפתחיה קרא בקובץ INFO שבתיקיה Terra.

התקנת Terra

מומלץ להסיר גירסה קודמת של Terra, אם קיימת.

1. הכנס את התקליטור לכונן.
2. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**, לחץ על **עיון**.
3. בחר בכונן התקליטורים בתיקיה **Terra** ובקובץ בשם **SETUP.EXE**.
4. לחץ על לחצן **פתח**. בחר בלחצן **אישור**.
5. פעל לפי ההוראות על המסך לפי הסדר (מימין לשמאל):

   Next, Yes, Next, Next, Next, Next, Finish.

ייתכן שבמהלך ההתקנה תתבקש להתקין רכיבי DirectX. אם במחשב שלך מותקנת מערכת ההפעלה Windows 95 הפועלת עם ממשק עברית (לחצן **התחל**), יהיה עליך לשנות את ההגדרות האזוריות:

1. בחר בלחצן **התחל**, **הגדרות**, **לוח הבקרה**, **הגדרות אזוריות**.
2. במקום **עברית** בחר **אנגלית** (ארצות הברית).
3. בחר בלחצן **החל**.
4. לשאלה האם ברצונך לאתחל את המחשב מחדש? ענה **כן**.
5. עתה, יהיה עליך להתחיל את התקנת Terra מחדש.

הפעלת Terra

לחץ על לחצן התחל, תוכניות, Terra, TerraViewer.

פעולה	מקשים
הגבר מהירות	Shift+A
האט מהירות	Shift+Z
תנועה מעלה/מטה/ימינה/שמאלה	מקשי חיצים

הוראות הפעלה מפורטות נמצאות בתפריט Help שבתוכנת Terra.

התקליטור חייב להיות בכונן בעת הפעלת התוכנה.



טיפ!

תוכל להאט את המהירות בעזרת Shift+Z לא רק למהירות אפס, אלא מתחת לזה. המשמעות היא ש... תטוס אחורה!!! שווה בדיקה!!!

התקנת הדפדפן
Microsoft Internet Explorer 5

תוכנית ההתקנה מזהה את גרסת מערכת ההפעלה ומתקינה את גרסת הדפדפן הדרושה.



1. הכנס את התקליטור לכונן.
2. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
3. לחץ על לחצן **עיון**.
4. בחר בכונן התקליטורים בתיקיה **Software\IE5** ובקובץ בשם **Setup.exe**.
5. לחץ על לחצן **פתח**.
6. לחץ על לחצן **אישור**.
7. פעל לפי ההוראות על המסך.

NETEX

במקום לרשום **http://www.hod-ami.co.il** פשוט רישמו **הוד-עמי** והנה אתם באתר ההוצאה. פרטים בהמשך.

במטרה להגיע לאתר מסוים באינטרנט, שכתובתו אינה ידועה, אנו משתמשים בדרך-כלל באחת משתי דרכים:

- ניחוש של כתובת האתר
- פנייה לאינדקס או למנוע חיפוש

שתי הפעולות הן מסורבלות וגוזלות זמן ואנרגיה מיותרים. ניחוש הכתובת מחייב הקלדה של הכתובת המלאה באנגלית בדיוק מושלם, והוא עשוי להיות הליך גוזל זמן, במיוחד כאשר לא מצליחים למצוא את האתר בניסיון ראשון או בכלל.

עם netex לא צריך לנחש כתובות או לגלוש למנועי חיפוש כדי להגיע לאתר מסוים ברשת! פשוט מקלידים את שם האתר בעברית בחלון הכתובת בדפדפן, ומגיעים אליו ישירות.

אפשרויות השימוש במערכת

גלישה ישירה לאתר לפי שמו או לפי נתונים הקשורים בו. מקלידים בחלון הכתובת של הדפדפן שם של אתר או חברה או מילות מפתח הקשורות באתר (בכל סדר שהוא), ומגיעים אליו ישירות.

לדוגמה:

- מקלידים הוד-עמי או הוצאת הוד-עמי או ספרי מחשבים ומגיעים ישירות לאתר הוצאת הוד-עמי לספרי מחשבים.
- מקלידים בנק דיסקונט - ומגיעים ישירות לאתר של בנק דיסקונט.
- מקלידים בנק - ומגיעים לדף ובו רשימה של קישורים לאתרים שקוטלגו תחת המילה בנק.
- מקלידים סלקום או 052 - ומגיעים ישירות לאתר של חברת סלקום;
- מקלידים עכבר העיר - ומגיעים ישירות לאתר של העכבר;
- מקלידים 144 - ומגיעים למודיעין 144 של בזק;
- מקלידים דפי זהב, זהב דפי, מדריך דפי וכו' - ומגיעים ישירות לאתר של מדריך דפי זהב.

התחברות למערכת הניתוב החדשה של ישראל

כל שעליכם לעשות כדי להתחבר ל-NETEX הוא להתקין תוכנה קלה וחכמה:

1. יש לסגור את כל התוכנות הפתוחות כולל הדפדפנים הפועלים.
2. מתוך **סייר Windows** הפעילו את הקובץ **netex100.exe** שבתיקיה **X:\Software\NetEx**.
3. פעלו בהתאם להוראות.

בזמן ההתקנה התוכנה מזהה את הדפדפן/ים שמותקנים במחשב, והיא תפעל עם כולם. מייד אחרי סיום ההתקנה תוכלו להתחיל לגלוש חכם ובעברית.

המערכת שקופה למשתמש, כלומר היא "מתלבשת" על הדפדפן הרגיל ואינה נראית כלל. אין צורך להפעיל אותה או לבצע פעולה כלשהי כדי להשתמש בה: פשוט מקלידים את שם האתר המבוקש בשדה הכתובת של הדפדפן, ומגיעים ישר אליו.

התוכנה אינה מפריעה לעבודה רגילה עם הדפדפן. היא נכנסת לפעולה רק כאשר מקלידים נתון שאינו כתובת אינטרנט רגילה (URL). כאשר תקלידו **www.hod-ami.co.il** תגלשו ישירות לאתר **הוצאת הוד-עמי**, בדיוק כפי שנהגתם לגלוש לפני התקנת התוכנה (ללא מעורבות המערכת). אך אם תרצו, תוכלו להקליד **הוד-עמי** בשדה כתובת ולהגיע במהירות לאותו אתר בדיוק.

היכן נמצאים הקבצים הקשורים לספר זה?

התיקיה הרלוונטית לספר זה: **59240**

תחת תיקיה **X:\Books\59240** תמצא קובץ ו-3 תיקיות משנה:

- תיקיה בשם **Icons** בה נמצאים קבצי ico. בהם תשתמש בפיתוח הפרויקטים השונים. העתק תיקיה זו לכונן הקשיח אצלך.
- תיקיה בשם **Images** בה נמצאים קבצי bmp. תמונות בהן תשתמש בפיתוח הפרויקטים השונים. העתק תיקיה זו לכונן הקשיח אצלך.
- תיקיה בשם **VBProjects** המחולקת לעשר תיקיות נוספות בשם **Part X** (Part 1, Part 2, Part 3 ...), תיקיה לכל פרק. בתיקיות אלו תמצא את הפרויקים המוכנים. אין צורך להעתיקם לדיסק.
- קובץ גופן בשם **wingdng2.ttf** המשמש בפרויקטים מסוימים. להתקנתו, פתח את **לוח הבקרה**, בחר **גופנים**. בחלון שייפתח פתח את תפריט **קובץ** ובחר **התקנת גופן חדש**. בתיבת הדו-שיח שתיפתח בחר את כונן התקליטורים, בחר את התיקיה (**X:\Books\59240**) בחר את הגופן ולחץ **אישור**.

אנו מבקשים להתייחס **רק** לקבצים הנמצאים בתיקיה הרלוונטית לספר. בתיקיות אחרות נמצאים קבצים הרלוונטיים לספרים אחרים של ההוצאה.

העתקת התיקיות לדיסק

כדי להעתיק את התיקיות לדיסק:

1. לחץ על לחצן **התחל**, **תוכניות**, **סייר Windows**.
2. בכונן הקשיח שלך צור תיקיה חדשה בשם **VBProjects**.
3. הצג את תוכן תיקיה **Books** אשר בתקליטור.
4. סמן את התיקיות הרצויות וגרור אותן לתיקיה שיצרת.

מעתה, בעת התייחסות בגוף הספר לתיקיות שבתקליטור, עבור לתיקיה שיצרת.

מכיון שמקור הקבצים הוא התקליטור, הם מסומנים לקריאה בלבד. רצוי לשנות מאפיין זה בדרך זו:

1. לחץ על לחצן **התחל**, **תוכניות**, **הפנייה ל-MS-DOS**.
2. בחלון DOS רשום: **attrib -r +a c:\ VBProjects *.* /s**.
3. אם העתקת את התיקיות לתיקיה אחרת, רשום זאת במקום C:**VBProjects**.

אפשרות אחרת:

1. דרך הסייר היכנס לתיקיה בדיסק, שבה נמצאים הקבצים שהעתקת.
2. סמן קובץ מסוים או את כולם על ידי Ctrl+A.
3. הצב את סמן העכבר מעל האזור המסומן ולחץ לחיצה ימנית בעכבר.
4. מהתפריט המקוצר בחר **מאפיינים**.
5. בטל את הסימון בתיבה **קריאה בלבד** (דאג שתיבה זו תהיה ריקה).
6. לחץ על **החל**.
7. לחץ על **אישור**.

הערה!

בהחלט ייתכן שבתיקיה Software תמצא מספר תוכנות המוזכרות בספר, או שתהיינה שימושיות עבורך, למשל, תוכנת WinAmp להשמעת קבצי MP3 או MapEdit ליצירת מפת תמונה וכדומה.

התקנת Visual Basic 6 Working Model

על התקליטור שמצורף לספר, נמצאים קבצי ההתקנה של Microsoft Visual Basic 6.0 (Working Model). גירסה זו אמנם אינה מלאה, אך שימושית בהחלט.

התקנת Visual Basic 6 Working Model דורשת כי דפדפן האינטרנט Internet Explorer 4 יהיה מותקן במחשב. לצורך התקנת Visual Basic 6 Working Model עליך **להסיר** את התקנת Internet Explorer גירסה 3, אם קיימת, ורק אז לבצע את התקנת Visual Basic 6 Working Model.

הערה!

התקנת התוכנה נבדקה בהוצאת **הוד-עמי** על מחשב שיש בו Internet Explorer 4. ההתקנה לא נבדקה על מחשב שיש בו גירסה חדשה יותר.

להתקנת Internet Explorer 4 (גירסה מלאה הכוללת תמיכה מלאה בעברית):

1. לחץ על **התחל**.
2. בחר **הפעלה**.
3. הקלד את הנתיב X:\SoftWare\IE401SR1\I386\Setup (כאשר X היא האות המייצגת את כונן התקליטורים במחשב שלך).
4. לחץ על **אישור** ועקוב אחר ההוראות המופיעות על המסך.
5. בסיום ההתקנה תתבקש לאתחל את המחשב מחדש. עשה זאת.

לתשומת לבך!

אם במחשב היתה מותקנת גירסה קודמת של Internet Explorer, במהלך ההתקנה צפויה להופיע תיבת דו-שיח המורה על התנגשות בין גרסאות. לשאלה האם ברצונך לשמור את הקובץ הקיים במחשבך השב **כן** וההתקנה תמשיך כרגיל.

לאחר התקנת Internet Explorer 4 ניתן להתקין את Visual Basic 6 Working Model באופן הבא:

1. לחץ על **התחל**.
2. בחר **הפעלה**.
3. לחץ על **עיון.**
4. עבור לכונן התקליטורים.
5. פתח את התיקיה **SoftWare**.
6. פתח את התיקיה **Visual Basic 6 Working Model**.
7. בחר בקובץ **Setup.exe** ולחץ על **פתח**.
8. בתיבת הדו-שיח **הפעלה** לחץ על **אישור** וההתקנה תחל.
9. במסך הראשון: **Visual Basic 6 Working Model** לחץ על **Next**.
10. במסך השני: **End User License Agreement** בחר: **I accept the agreement** ולחץ על **Next**.
11. במסך השלישי: **Product Number and user ID** הקלד את שמך ולחץ על **Next**.
12. במסך **Install DCOM98** ודא שתיבת הסימון מסומנת ולחץ על **Next**.
13. לאחר התקנת הרכיב **DCOM98** תוכנית ההתקנה תודיע שכעת יש לאתחל את המחשב מחדש לחץ על **OK**.
14. לאחר שהמחשב יעלה מחדש, תימשך ההתקנה. המסך הבא שיופיע הוא **Choose Common Install Folder**. בדוק אם המיקום והנתיב שמציעה תוכנית ההתקנה מתאימים לך. אם לא, לחץ על **Browse** ובחר מיקום אחר. כעת כשהמיקום והנתיב מתאימים, לחץ על **Next**.
15. במסך: **Visual Basic 6 Working Model Setup** תופיע הודעה שיש לסגור כל יישום פתוח (למעט תוכנית ההתקנה), עשה זאת (באמצעות לחיצה על לחצן היישום שבשורת המשימות) ולחץ על **Continue**.
16. בשלב הבא לחץ על **OK** והמתן בעת שתוכנית ההתקנה מחפשת רכיבים מותקנים.
17. במסך הבא שיופיע בחר בסוג ההתקנה. אם יש לך מספיק מקום בדיסק שבחרת בחר בהתקנת **Typical**.
18. אם תוכנית ההתקנה תבקש להחליף Dll ענה **No** ואחר כשתשאל אם ברצונך להשאיר את הקבצים כמות שהם ענה **Yes**.
19. בשלב זה תתבקש לאתחל את Windows מחדש, לחץ על **Restart Windows**.
20. כשהמחשב "יעלה" מחדש תופיע תיבת דו-שיח המאפשרת רישום עותק התוכנה באמצעות האינטרנט. פעל לפי ראות עיניך.

תיקיה ראשית SoftWare (רשימה חלקית)

שם תוכנה	תיאור	קובץ הפעלה	מבצע
Adobe Acrobat	תוכנה לצפייה בקבצי pdf	Ar32e301.exe	התקנה
Adobe Photoshop 5 ME	תוכנה לעיבוד תמונה	Setup.exe	התקנה
Babylon	תוכנה לתרגום ולבדיקת איות	B30502h3.exe	התקנה
ClnSys	מחיקת קבצי dll שאין צורך בהם (מאמר בנושא הופיע בחושבים חלונות גיליון 22)	ClnSys16.exe	התקנה
FontsPekan	גופנים בעברית	FontsPekan.exe	פריסה
Gohst 5.1c	תוכנה להעתקת תוכן כוננים קשיחים	g51c_trl.exe	פריסה
ICQ	תוכנה לתקשורת אישית באינטרנט	ICQ99a.exe	התקנה
Paint Shop Pro 5.01	תוכנה ליצירה, לעיצוב ולעיבוד תמונות	Psp501ev.exe	התקנה
Power Toys	תוכניות שירות עבור Windows 95	PowerToy.exe	פריסה
WinAmp	תוכנה להשמעת קבצי MP3 (מוסיקה)	WinAmp.exe	התקנה
WinZip	תוכנית לפריסה ולדחיסה של קבצים	WinZip7sr1.exe	התקנה
WordView	תוכנית לצפייה בקבצי doc	WordView.exe	התקנה

נספח: התקליטור המצורף

בתקליטור המצורף לספר זה תוכל למצוא מספר דברים:

- **קטלוג CD** - קטלוג ספרי המחשבים הממוחשב האינטראקטיבי של **הוצאת הוד-עמי** (נדרשת התקנת תוכנה; כולל חיפוש, הדפסה, דפדוף).
- **קטלוג HTML** - קטלוג ספרי המחשבים האינטראקטיבי של **הוצאת הוד-עמי** (**לא** נדרשת התקנת תוכנה; מומלץ לצפייה עם IE מגירסה 4 ומעלה).
- **ספרים לדוגמה** בהם ניתן לעיין וגם להדפיס.
- תוכנת **מבחן אישי** - לבדיקת ידע בתוכנות Office ובה מאגר של מעל 400 שאלות.
- גיליון מלא של ירחון **חושבים חלונות** ובו מאמרים ב- Windows 95, Word 97 ו-Excel 97.
- מספר תוכנות עזר שימושיות.
- קבצי תרגול.

אם מנהל התקן כונן התקליטורים המותקן הוא 16 סיביות (וזה יכול להיות גם אם הכונן חדש) - ייתכן שתראה רק את 8 התווים הראשונים של שם הקובץ (במקרה שהמקור ארוך יותר), או ייתכן שתראה שהתיקיות ריקות.

הטוב ביותר הוא להתקין מנהל התקן 32 סיביות, או לקנות כונן תקליטורים חדש ולוודא שמצורף אליו מנהל התקן 32 סיביות.

קטלוג CD - הקטלוג הצבעוני האינטראקטיבי של הוד-עמי

הוצאת הוד-עמי גאה לבשר על הקטלוג הצבעוני האינטראקטיבי היחידי בארץ בתוכנה!

הקטלוג **הצבעוני** האינטראקטיבי של הוד-עמי עושה שימוש בטכנולוגיית IBL (Internet/Intranet Based Learning), שהינה המילה האחרונה בכל הקשור לשימושים ולאפשרויות חדשות הגלומות בשימוש בטכנולוגיות אינטרנט.

בעזרת הקטלוג הצבעוני האינטראקטיבי של הוד-עמי תוכל:

- לעיין במידע על ספרי ההוצאה מתי שתרצה (לאחר התקנת התוכנה, בחר **התחל**, **תוכניות**, **קטלוג הוד-עמי**).
- לעבור במהירות ובקלות בין הקטלוג הצבעוני האינטראקטיבי של הוד-עמי והיישום בו אתה עובד.
- לדפדף בקטלוג **הצבעוני** האינטראקטיבי ספר אחר ספר, או לפי נושאים וקבוצות.
- להדפיס את המחירון המלא ומידע על כל ספר.
- לגשת במהירות, בגישה אינטואיטיבית, תוך התמקדות מהירה בספר המבוקש.
- לעיין בקטלוג **הצבעוני** האינטראקטיבי בקצב אישי שלך.
- לנווט את דרכך בקטלוג **הצבעוני** האינטראקטיבי ולחזור ולהתרענן בכל נושא בכל רגע.
- להוריד עדכונים אודות ספרים חדשים מהאינטרנט בכתובת:

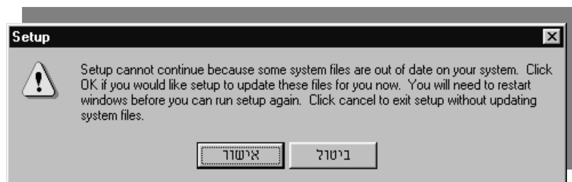
http://www.hod-ami.co.il/

הקטלוג מתאים לעבודה ב-Windows 95/98 עם Internet Explorer 3.02a בגירסה העברית, אך עם זאת, ניתן לצפייה גם עם דפדפן MS-IE4 ומעלה.

הקטלוג מתאים לעבודה ב-Windows 98 אם כי ההתקנה דורשת התערבות מצידך כדי שתושלם. לפרטים קרא בקובץ README הנמצא בתיקיה X:\Catalog.

1. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד את הפקודה **X:\Catalog\Hod-ami** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.

3. עקוב אחר ההוראות המופיעות על המסך ופעל לפיהן. כאשר מופיעה תיבת ההודעה **Setup** המודיעה על התקנת רכיבים חדשים - לחץ **אישור**. פעולה זו מצריכה אתחול מחדש של המחשב.



3. לאחר שהמחשב עולה מחדש הפעל את תוכנית ההתקנה מחדש:

- לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
- בתיבת הטקסט הקלד את הפקודה **X:\Catalog\Hod-ami** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.

5. לחץ על לחצן **Setup**. מופיע חלון עם רקע כחול ובו הכותרת: **"התקנת קטלוג הוד-עמי"**. לחץ **המשך**.

6. בחלון **התקנת קטלוג הוד-עמי** לחץ על תמונת המחשב. אם מופיעה תיבת דו-שיח בה תתבקש להגדיר היכן להתקין את Internet Explorer, השתמש בברירת המחדל. לא תיבהל, זו אינה התקנה "אמיתית" של Internet Explorer, אלא רק של מספר רכיבים הדרושים להפעלת הקטלוג (שים לב, שאין צורך בהתקנה מלאה של Internet Explorer). לחץ **OK**.

6. בסיום ההתקנה כולה תופיע תיבת דו-שיח. לחץ על **אישור**. יופיע חלון **Setup**, אשר את אתחול המחשב פעם נוספת על ידי בחירה בלחצן **כן**.

7. בחר בתפריט **התחל** באפשרות **הפעלה**.

8. בתיבת הטקסט הקלד **X:\Catalog\Update9809.exe** (החלף את האות X באות הכונן המתאימה) ולחץ על **אישור**. זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.

9. קרא את ההודעה ולחץ **אישור**.

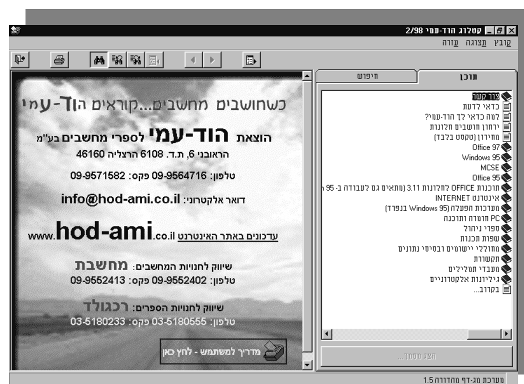
10. בחלון WinZip Self-Extractor בחר בלחצן **Unzip** והמתן.

11. קרא את ההודעה ולחץ **אישור**.

12. בחלון WinZip Self-Extractor בחר בלחצן **Close**.

13. חזור על סעיפים 8-14 עבור קובץ **X:\Catalog\Update????.exe**. (ארבעת סימני השאלה הם עבור 9903, או מספר גדול יותר כמו 9904, 9905 וכדומה).

14. לאחר שהמחשב עולה מחדש, לחץ על לחצן **התחל**, **תוכניות** ובחר באפשרות **קטלוג הוד-עמי**.



בכל חודש הקטלוג מתעדכן. ניתן להוריד קובץ עדכון מאתר ההוצאה בכתובת:

www.**hod-ami**.co.il

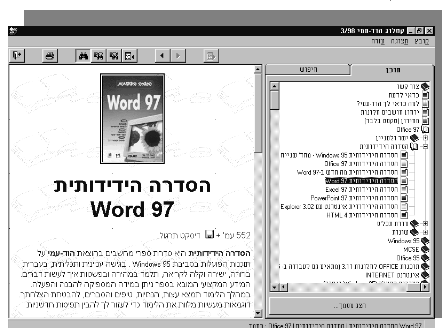
הפעלת התוכנה שהותקנה

להפעלת הקטלוג הצבעוני של הוצאת **הוד-עמי**, לחץ על **התחל**, עבור ל**תוכניות** ובתחתית התפריט תראה את **קטלוג הוד-עמי**. בחר בו.

> התקנת הקטלוג ב-Windows 95 עשויה (לא תמיד) לשבש את העבודה עם הכתבן. במקרה זה יש להעתיק את קובץ Riched32.dll לתיקיית המערכת של Windows 95. בדרך כלל, זו תיקיה C:\Windows\System..

מבנה המסך הראשי

בהפעלת התוכנה יופיע מסך ובו שלושה אזורים עיקריים:



- מימין יופיע **סייר המסמכים** (דומה ל**סייר Windows**). למעשה, זהו עץ המציג את פרטי הקטלוג השונים. אזור זה ישמש בהמשך גם לתהליך החיפוש.
- בצד שמאל יופיע דף הפתיחה. זהו **אזור התוכן**.
- בחלק העליון יופיעו שורת התפריט וסרגל הכלים.

כך נראה מסך של קטלוג **הוד-עמי** הנמצא בתקליטור המצורף לספר.

סיור מודרך בעזרים קלים

1. לחץ לחיצה כפולה על הסמל שלידו רשום **Office 97**.
2. לחץ לחיצה כפולה על הסמל שלידו רשום **הסדרה הידידותית/קוראים יודעים**.
3. תיפתח רשימת מסמכים הקשורים לנושא.
4. לחץ לחיצה כפולה על הסמל שלידו רשום **קוראים יודעים Word 97**.
5. בצד שמאל יוצג מסמך שכותרתו **קוראים יודעים Word 97**.
6. לחץ על הסמל שבסרגל הכלים שבראש החלון, כדי לעבור לדף הבא.
7. יופיע דף שכותרתו **הסדרה הידידותית הכרת המחשב האישי**.
8. גרור מטה את פס הגלילה של החלון השמאלי, כדי לקרוא את המשך התיאור.
9. אם ברשותך מדפסת, תוכל להדפיס את הדף על ידי לחיצה על הסמל .

שינוי מבנה המסך

ניתן לשנות את תצוגת המסמך על ידי שני לחצנים :



בהצבת הסמן בין אזור סייר המסמכים לאזור המסמך יוצג חץ דו-ראשי. בגרירה ימינה ושמאלה תוכל לשנות את חלוקת המסך בין שני האזורים.



הצג/הסתר סייר מסמכים. בלחיצה על לחצן זה, אזור המסמך "ישתלט" על כל המסך. לחיצה נוספת על לחצן זה תחזיר את המצב לקדמותו.

סרגל הכלים



יציאה מהתוכנה.



הדפס מסמך. המסמך המוצג על המסך יישלח להדפסה.



הצג/הסתר סייר מסמכים. בלחיצה על לחצן זה, אזור המסמך "ישתלט" על כל המסך. לחיצה נוספת על לחצן זה תחזיר את המצב לקדמותו.



הקודם/הבא בסייר המסמכים. בהתאם למיקום הסמן יוצג המסמך הקודם/הבא בעץ המסמכים (ראה פירוט בהמשך).



הצג מיקום המסמך בתוכן. הסמן יתמקם בסייר המסמכים על השורה המתאימה למסמך הנוכחי. שים לב, ייתכן שהסמן בעץ המסמכים ניצב על מסמך שאינו המסמך המופיע באזור התוכן באותו שלב (כדי להתאימם נשתמש בכלי הבא).


מסמך קודם/הבא. דפדוף קדימה ואחורה במסמכים שכבר עברנו עליהם בהפעלה נוכחית (ראה פירוט בהמשך).


הצג תת-עץ במסמך. כשהמסמך נייצב על נושא ובו מסמכים ו/או תת-נושאים נוספים, שימוש בלחצן זה באזור סייר המסמכים יפרוס את תת-הנושאים והמסמכים ובצד השני יוצג מסמך אחד ארוך המכיל את כל המסמכים.

דפדוף בעץ המסמכים

ניתן לבצע את פעולת הדפדוף בשלוש דרכים שונות:

 מעבר על עץ המסמכים.

 פתיחת נושא.

 סגירת פרק.

מעבר על עץ המסמכים

 נושא כללי.

 נושא או תת-נושא פתוח.

 ספר.

פתיחת נושא

יש שלוש אפשרויות לפתיחת נושא:

 לחיצה כפולה על שם הנושא.

 לחיצה על סימן "+" המופיע משמאל לסמל הנושא.

 על ידי מקש החיצים: →.

סגירת פרק

יש שלוש אפשרויות לסגירת פרק:

 לחיצה כפולה על שם הנושא.

 לחיצה על סימן "-" המופיע משמאל לסמל הנושא.

 על ידי מקש החיצים: ←.

מעבר בין צוהי עץ המסמכים במגירת לוח המקשים

- לחיצה על אות כלשהי תעביר את הסמן לשורה הבאה בעץ, המתחילה באותה אות (אם אין שורה כזו, לא יקרה דבר).
- לחיצה על ↓ תעביר את הסמן לשורה הבאה ולחיצה על ↑ תעביר את הסמן לשורה הקודמת בעץ.

הצגת מסמך

יש שתי אפשרויות להצגת מסמך:

- כאשר הסמן ניצב על שם המסמך - לחיצה על לחצן [הצג מסמך...].
- לחיצה כפולה על שם המסמך בעץ.

שימוש בלחצנים בסרגל הכלים

דפדוף בעזרת לחצנים אלה דומה לדפדוף בספר, דף אחר דף, קדימה או אחורה. לחצנים אלה משמשים למעבר בין המסמכים, קדימה ואחורה, על פי סדר הופעתם בעץ התפריטים (נקודת ההתחלה היא מיקום הסמן בעץ). אם מגיעים לפרק או לתת-נושא סגור - הוא ייפתח אוטומטית ויוצג המסמך הראשון.

כל לחיצה תפתח את המסמך המתאים (בחלק השמאלי) ותעביר את הסמן לשורה המתאימה בעץ (בחלק הימני).

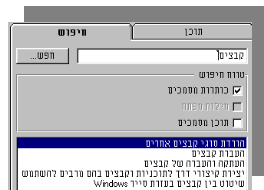
שימוש בלחצנים בסרגל הכלים

דפדוף בין מסמכים שהופעלו בהפעלה הנוכחית. המסמך הנוכחי הינו המסמך האחרון ממנו אפשר לעבור אחורה עד למסמך הראשון של ההפעלה (מסך הפתיחה). בדרך בין המסמך הראשון למסמך הנוכחי תוכל לדפדף קדימה ואחורה.

הורדת עדכונים

הקטלוג הצבעוני מתעדכן מדי חודש. את העדכון ניתן להוריד מהאינטרנט (חינם!) מאתר ההוצאה. בפתיחת הקטלוג בצד שמאל כתוב **עדכונים באתר האינטרנט**. אם אתה מחובר לאינטרנט, הצב את הסמן על כתובת זו ולחץ. להרחבת התצוגה הקה לחץ על **משקפת**. באתר **הוד-עמי** בחר בקישור **קטלוג והורדת עדכון חודש**. הורד את העדכון, וזכור היכן אתה שומר את הקובץ. צא מהקטלוג ונתק את החיבור לאינטרנט. הפעל את **סייר Windows** ואתר את הקובץ. לחץ עליו לחיצה כפולה. קרא את ההודעה ולחץ **אישור**. לחץ **UnZip**, קרא את ההודעה ולחץ **אישור**. לחץ **Close** והפעל מחדש את הקטלוג. זהו, הקטלוג מעודכן!!!

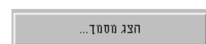
חיפוש



1. עבור לכרטיסיה **חיפוש** בחלק הימני של החלון.
2. הקלד רצף תווים. זו יכולה להיות מילה, או חלק ממילה.
3. קבע את טווח החיפוש - האם לחפש בכותרות המסמכים ו/או בתוכן המסמכים.
4. הקש **Enter** או לחץ על ![חפש...]. אם נמצאו מסמכים, תופענה כותרותיהם.

הצגת המסמך

יש שתי אפשרויות להצגת מסמך:

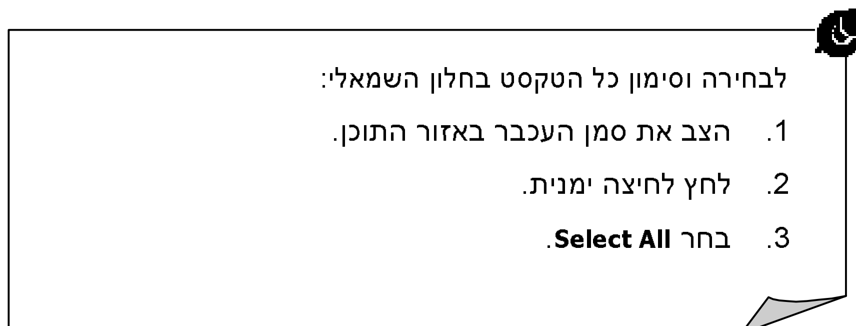
- כאשר הסמן ניצב על שם המסמך - לחיצה על לחצן .
- לחיצה כפולה על שם המסמך.

שים לב! במסמך שמוצג מחלון החיפוש, הטקסט שחיפשת יודגש בצבע אדום.

העתקת טקסט (ללא תמונות) לתוכנות אחרות

אפשר לסמן קטע ממסמך, להעתיקו ולהדביקו במקום אחר (למשל, במסמך Word).

1. סמן את הטקסט שברצונך להעתיק (הסימון מתבצע על ידי לחיצה וגרירת העכבר על האזור שברצונך להעתיק). הטקסט שסומן ייצבע בצבע כחול.



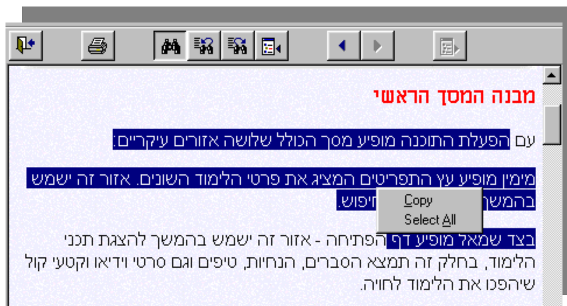
לבחירה וסימון כל הטקסט בחלון השמאלי:

1. הצב את סמן העכבר באזור התוכן.
2. לחץ לחיצה ימנית.
3. בחר **Select All**.

2. ביצוע העתקה:

- הצב את סמן העכבר על הקטע המסומן.
- לחץ לחיצה ימנית ובחר **Copy**, או לחץ **Ctrl+C**.
- עבור לתוכנה אחרת, כמו Word, פנקס רשימות, כתבן, וכדומה.

- הצב את נקודת הכניסה במקום שבו תרצה להוסיף את הטקסט ובצע אחת מהפעולות הבאות: בחר בתפריט **עריכה**, **הדבק**, או לחץ על לחצן **הדבק** , או הקש **Ctrl+V**, או לחץ לחיצה ימנית ובתפריט המקוצר בחר **הדבק**.



העתקת תמונה לתוכנות אחרות

אפשר להעתיק תמונה ממסמך שמופיע על המסך ולהדביקה במקום אחר (למשל במסמך Word).

1. הצב את הסמן על התמונה שברצונך להעתיק.
2. לחץ לחיצה ימנית ובתפריט המקוצר בחר **Copy**.
3. עבור לתוכנה אחרת: תוכנה גרפית, Word וכדומה.
4. לביצוע ההדבקה, בחר בתפריט **עריכה**, **הדבק**;

   או לחץ על לחצן **הדבק** ;

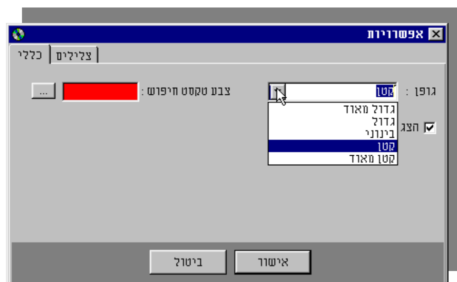
   או הקש **Ctrl+V;**

   או לחץ לחיצה ימנית ובתפריט המקוצר בחר **הדבק**.

אפשרויות תצוגה

בשורת התפריט בחר **תצוגה**, **אפשרויות**. בחלון זה ניתן לשנות את המאפיינים הבאים:

- גודל הגופן המוצג במסמכים השונים.
- צבע טקסט החיפוש (ברירת המחדל - אדום).
- האפשרות להציג שורת מצב, או לא להציגה.
- הצללים שמשמעם התוכנה כאשר מתחלפים המסמכים.



בתיבת הדו-שיח **אפשרויות**, תוכל לקבוע גם את צבע טקסט החיפוש.

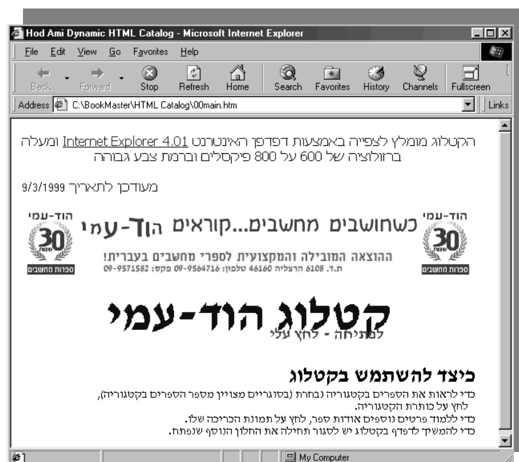
קטלוג HTML - קטלוג צבעוני

הוצאת הוד-עמי גאה לבשר על קטלוג HTML **צבעוני** העושה שימוש בטכנולוגיית DHTML (Dynamic HTML) שהינה המילה האחרונה בעיצוב דפי Web באינטרנט.

בעזרת קטלוג HTML **צבעוני** תוכל:

- לעיין במידע על ספרי ההוצאה מתי שתרצה (לחיצה כפולה... וזהו!).
- לעבור במהירות ובקלות בין הקטלוג **הצבעוני** והיישום בו אתה עובד.
- לעיין במידע על כל ספר וספר.
- לגשת במהירות, בגישה אינטואיטיבית, תוך התמקדות מהירה בספר המבוקש.
- לעיין בקטלוג **הצבעוני** בקצב אישי שלך.
- לנווט את דרכך בקטלוג **הצבעוני** ולחזור ולהתרענן בכל נושא בכל רגע.

הקטלוג מומלץ לצפייה בעזרת Internet Explorer מגירסה 4 ומעלה.



1. פתח את סייר Windows.
2. עבור לתיקיה HTML Catalog אשר בתקליטור המצורף.
3. אתר את הקובץ main.htm00 והפעל אותו (בדרך כלל על ידי לחיצה כפולה).
4. קרא את ההוראות כיצד להשתמש בקטלוג שבמסמך פתיחה זה.
5. לחץ על הכותרת **קטלוג הוד-עמי**.
6. יוצגו הנושאים השונים: Office 97, Windows 98 וכדומה.



7. הצב את הסמן על כותרת הנושא, למשל, **Office 97** ולחץ בעכבר.

8. יוצגו הספרים השייכים לאותו נושא.



9. לחץ בעכבר על הספר לגביו אתה מעוניין במידע (שים לב, הסמן **לא** ישתנה).

10. ייפתח חלון בו תוכל לקרוא מידע אודות הספר (מה הוא מכיל, למי הוא מיועד, מה נמצא בתקליטור המצורף, מי המחבר, כמה עמודים בספר וכדומה).

כדי להמשיך לעיין בקטלוג, עליך **לסגור** את החלון המכיל את פרטי הספר.

המחירון המעודכן של ספרי ההוצאה נמצא באתר האינטרנט:

www.hod-ami.co.il

התקנת קטלוג HTML

מי שרוצה להפעיל את קטלוג HTML מתוך התקליטור, צריך לנהוג לפי ההוראות הרשומות לעיל. הפעלה מתוך התקליטור מחייבת את המצאות התקליטור בכונן. בהתקנה פשוטה ניתן להעתיק את הקטלוג לדיסק הקשיח. להלן ההוראות להתקנת קטלוג HTML:

1. לחץ על **התחל** ובחר **הפעלה**.
2. בעזרת לחצן עיון סמן את הקובץ **HodAmiHTMLcatalog.EXE** אשר בתיקיה **X:\HTML Catalog**.
3. לחץ **פתח**.
4. לחץ **אישור** פעמיים.
5. לחץ על לחצן **Unzip.**
6. לחץ אישור.
7. על שולחן העבודה מופיע סמל עם כיתוב קטלוג ספרי מחשבים בהוצאת הוד-עמי.
8. הפעל אותו.



קטלוג ספרי מחשבים בהוצאת הוד-עמי

Acrobat Reader - התקנה

תוכנה זו יש להתקין, כדי לקרוא את דוגמת הירחון "**חושבים חלונות**" ואת **הספרים לדוגמה** וגם כדי להפעיל את תוכנות הבחינה לדוגמה המצורפות לתקליטור זה. התוכנה פועלת במערכות הפעלה **Windows 95/98 בלבד!**

1. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד את הפקודה **X:\Software\Adobe Acrobat\Ar32e301** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.
3. בחלון הבא בחר בלחצן **כן** להמשך ההתקנה.

4. אשף ההתקנה מתקין את הרכיבים הנדרשים. עליך ללחוץ על **Next**, **Yes** ו-**Next** פעם נוספת, כדי לסיים את ההתקנה.
5. בסיום ההתקנה לחץ על **Finish**.
6. קרא את הודעת סיום ההתקנה שמופיעה ולחץ **אישור**.

ספרים לדוגמה

בתיקיה **Books DEMO** תמצא מספר דוגמאות מספרי ההוצאה. כדי לראות וגם להדפיס את הקבצים (פורמט pdf), עליך להתקין את Acrobat Reader :

1. פתח את תפריט **התחל**, **תוכניות**, **Adobe Acrobat**, **Acrobat Reader 3.01**.
2. פתח את תפריט **File** ובחר באפשרות **Open**.
3. בתיבה **חפש ב** : בחר בכונן התקליטורים שלך ועבור לתיקיה **Books DEMO**.
4. בחר בקובץ ולחץ על לחצן **Open**.

חושבים חלונות - הפעלה

1. פתח את תפריט **התחל**, **תוכניות**, **Adobe Acrobat**, **Acrobat Reader 3.01**.
2. פתח את תפריט **File** ובחר באפשרות **Open**.
3. בתיבה **חפש ב** : בחר בכונן התקליטורים שלך ועבור לתיקיה **ThinkWin**.
4. בחר בקובץ **ThinkWin - Vol 5.pdf** המופיע בתיקיה, כדי לקרוא גיליון 5 במלואו.
5. בחר בקובץ **ThinkWin - Brief.pdf** המופיע בתיקיה זו, כדי לקרוא את תקצירי הגליונות הקודמים.

ומה הלאה...

הזמנת מנוי - חושבים חלונות **או פקס 09-9571582**

שלח בדואר לפקודת **הוד-עמי ת.ד. 6108 הרצליה 46160**

סמן/י X לפי בחירתך:

אני מעוניין/ת לעשות מנוי על הירחון **חושבים חלונות**.

☐ 12 גיליונות במחיר 99₪ כולל מע"מ

חובה למלא פרטים

שם משפחה ____________ שם פרטי ____________

כתובת מלאה ______________________

ישוב ______________________

מיקוד ____________

טלפון ____________ פקס ____________

e-mail: ______________________

ויזה - 16 תשלומים   דיינרס - 14 תשלומים

ישראכרט - 8 תשלומים   מאסטרקארד - 16 תשלומים

מספר כרטיס אשראי

בתוקף עד ___/___   תעודת זיהוי

כתובת למשלוח החשבוניות (במידה והיא שונה מהכתובות שלעיל):

____

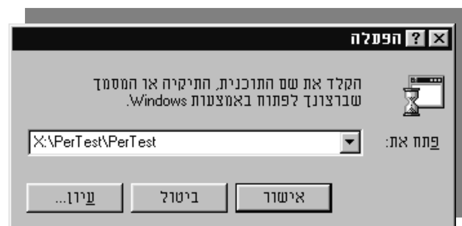
____

אישור: תאריך   חתימת המזמין

מבחן אישי - התקנה

מבחן אישי מאפשר לך לבחון את ידיעותיך בתוכנות Office השונות. בתקליטור המצורף לספר זה תמצא מספר מבחנים לתוכנת Word בגרסאות 7 ו-97.

1. בחר בתפריט **התחל** באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד **X:\PerTest\Pertest.exe** (החלף את האות X באות הכונן המתאימה) ולחץ על **אישור**.



זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.

3. קרא את ההודעה ולחץ **אישור**.
4. בחלון WinZip Self-Extractor בחר בלחצן **Unzip** והמתן.
5. לחץ **אישור**. תהליך ההתקנה מתחיל. המתן.
6. מופיעה הודעה שההתקנה הסתיימה בהצלחה. לחץ **אישור**.



התקנת ערכת מבחנים מלאה ל-Word 7/97

1. בחר בתפריט **התחל** באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד **X:\PerTest\PTWord.exe** (החלף את האות X באות הכונן מתאימה) ולחץ על **אישור**. זהו קובץ הנפרש אוטומטית וגם מותקן אוטומטית.
3. קרא את ההודעה ולחץ **אישור**.
4. בחלון WinZip Self-Extractor בחר בלחצן Unzip והמתן.
5. קרא את ההודעה ולחץ **אישור**.
6. בחלון WinZip Self-Extractor בחר בלחצן Close.

הפעלה - מבחן אישי הוד-עמי

לחץ על לחצן **התחל**, **תוכניות**, **מבחן אישי - הוד עמי**.

מה עוד בתקליטור?

הוצאת **הוד-עמי** מפיצה תוכנות אלו כבונוס ללקוחות ההוצאה ואינה מתיימרת לגבות תשלום עבור התוכניות המצורפות ו/או לתמוך בהם.



אזהרה!

השימוש בתקליטור זה הוא על אחריותו הבלעדית של המשתמש. המוצרים המותקנים בתקליטור זה מסופקים באחריות החברות המייצרות אותם. הוצאת **הוד-עמי** אינה אחראית, בכל צורה שהיא, לאופן ולטיב התוכנות המותקנות.

בכל שאלה לגבי תוכנה הנמצאת בתקליטור, יש לפנות למפתחי התוכנה (כל תוכנה בנפרד), כפי שמצוין בקבצי העזרה של התוכנה המדוברת.

הקבצים הם גרסאות **שיתופיות** (ShareWare) ו**חופשיות** (FreeWare).

גירסה **שיתופית** (ShareWare) מאפשרת לך, המשתמש, לבדוק את יעילות התוכנה ואת תאימותה לעבודה אותה אתה מבצע. אם נמצאה התוכנה מתאימה לצרכיך, עליך לשלם למפתחיה תשלום סמלי (לפי הרשום בקבצי העזרה של כל תוכנה ותוכנה בנפרד), כדי לקבל רישיון מלא לשימוש בה. רכישת רישיון לשימוש בתוכנה יפתח בפניך מיגוון אפשרויות שייתכן ולא עמדו לרשותך בהפעלת הגירסה השיתופית.

FontsPekan

קובץ זה יתקין במחשב שני גופנים בעברית לשימושך. בסיום ההתקנה יש לבצע את הפעולות הבאות:

1. לחץ על **התחל**, הצבע על **הגדרות** ובחר ב**לוח הבקרה**.
2. לחץ לחיצה כפולה על הסמל **גופנים**.
3. פתח את תפריט **קובץ** ובחר באפשרות **התקנת גופן חדש**.
4. עבור לתיקיה **C:\FontsPekan**.
5. לחץ על לחצן **בחר הכל** (סך הכל יש בתיקיה שני גופנים).
6. ודא שתיבת הסימון **העתק גופנים לתיקיית הגופנים** מסומנת. לחץ **אישור**.
7. סגור את חלון התיקיה **Fonts**.
8. סגור את חלון **לוח הבקרה**.

כעת, מוכנים הגופנים לשימוש בכל התוכנות המותקנות במחשב שלך: Word, Excel, PowerPoint וגם בתוכנות גרפיות, כגון Paint Shop Pro ו-PhotoShop.

הגופנים נקראים Tml-JUMP ו-Tml-step ויופיעו בתחתית רשימת שמות הגופנים (בדרך כלל). הרי דוגמה שלהם:

Tml-step

אבגדהוזחטיכךלמםנןסעפףצץקרשת1234567890

Tml-JUMP

אבגדהוזחטיכךלמםנןסעפףצץקרשת1234567890

Terra - מימד חדש בתצוגה - ים האלה ממטוס הציור

Terra היא משפחת מוצרים של חברת **סקיילייין מערכות תוכנה**, הבנויה סביב מנוע גרפי ייחודי ורב עוצמה, המאפשר תצוגה בזמן אמת של פני שטח בתלת מימד, ללא הגבלת פרטים וגודל, על גבי מחשבים ביתיים. מקור תמונת השטח הוא בתצלומי לווין ובתצלומי אוויר.

דרישות מערכת מינימליות :

- מעבד מסוג Pentium בטכנולוגיית MMX.
- זיכרון פנימי MB RAM32.
- זיכרון כרטיס מסך MB2.
- כונן תקליטורים במהירות x6.
- מערכת הפעלה Windows 95/98.
- התקליטור חייב להיות בכונן במהלך ההפעלה.

מנוע Terra לוקח את התצוגה בזמן אמת לרמות חדשות שלא נודעו בעולם מחשבי ה-PC. הוא מאפשר אין סוף מצבי תצוגה : החל ממבט קרוב לפרטי פרטים ועד תצוגה רחוקה עד האופק. מנוע Terra עושה שימוש בטכנולוגיית MMX החדשנית ומאפשר איכויות שבעבר היו נחלתם של מחשבים יקרים בלבד.

לפרטים נוספים על התוכנה ומפתחיה קרא בקובץ INFO שבתיקיה Terra.

התקנת Terra

מומלץ להסיר גירסה קודמת של Terra, אם קיימת.

1. הכנס את התקליטור לכונן.
2. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**, לחץ על **עיון**.
3. בחר בכונן התקליטורים בתיקיה **Terra** ובקובץ בשם **SETUP.EXE**.
4. לחץ על לחצן **פתח**. בחר בלחצן **אישור**.
5. פעל לפי ההוראות על המסך לפי הסדר (מימין לשמאל) :

   Next, Yes, Next, Next, Next, Next, Finish.

ייתכן שבמהלך ההתקנה תתבקש להתקין רכיבי DirectX. אם במחשב שלך מותקנת מערכת ההפעלה Windows 95 הפועלת עם ממשק עברית (לחצן **התחל**), יהיה עליך לשנות את ההגדרות האזוריות:

1. בחר בלחצן **התחל**, **הגדרות**, **לוח הבקרה**, **הגדרות אזוריות**.
2. במקום **עברית** בחר **אנגלית** (ארצות הברית).
3. בחר בלחצן **החל**.
4. לשאלה האם ברצונך לאתחל את המחשב מחדש? ענה **כן**.
5. עתה, יהיה עליך להתחיל את התקנת Terra מחדש.

הפעלת Terra

לחץ על לחצן התחל, תוכניות, Terra, TerraViewer.

פעולה	מקשים
הגבר מהירות	Shift+A
האט מהירות	Shift+Z
תנועה מעלה/מטה/ימינה/שמאלה	מקשי חיצים

הוראות הפעלה מפורטות נמצאות בתפריט Help שבתוכנת Terra.

התקליטור חייב להיות בכונן בעת הפעלת התוכנה.



טיפ!

תוכל להאט את המהירות בעזרת Shift+Z לא רק למהירות אפס, אלא מתחת לזה. המשמעות היא ש... תטוס אחורה!!! שווה בדיקה!!!

התקנת הדפדפן
Microsoft Internet Explorer 5

תוכנית ההתקנה מזהה את גרסת מערכת ההפעלה ומתקינה את גרסת הדפדפן הדרושה.



1. הכנס את התקליטור לכונן.
2. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
3. לחץ על לחצן **עיון**.
4. בחר בכונן התקליטורים בתיקיה **Software\IE5** ובקובץ בשם **Setup.exe**.
5. לחץ על לחצן **פתח**.
6. לחץ על לחצן **אישור**.
7. פעל לפי ההוראות על המסך.

NETEX

במקום לרשום **http://www.hod-ami.co.il** פשוט רישמו **הוד-עמי** והנה אתם באתר ההוצאה. פרטים בהמשך.

במטרה להגיע לאתר מסוים באינטרנט, שכתובתו אינה ידועה, אנו משתמשים בדרך-כלל באחת משתי דרכים:

- ניחוש של כתובת האתר
- פנייה לאינדקס או למנוע חיפוש

שתי הפעולות הן מסורבלות וגוזלות זמן ואנרגיה מיותרים. ניחוש הכתובת מחייב הקלדה של הכתובת המלאה באנגלית בדיוק מושלם, והוא עשוי להיות הליך גוזל זמן, במיוחד כאשר לא מצליחים למצוא את האתר בניסיון ראשון או בכלל.

עם netex לא צריך לנחש כתובות או לגלוש למנועי חיפוש כדי להגיע לאתר מסוים ברשת! פשוט מקלידים את שם האתר בעברית בחלון הכתובת בדפדפן, ומגיעים אליו ישירות.

אפשרויות השימוש במערכת

גלישה ישירה לאתר לפי שמו או לפי נתונים הקשורים בו. מקלידים בחלון הכתובת של הדפדפן שם של אתר או חברה או מילות מפתח הקשורות באתר (בכל סדר שהוא), ומגיעים אליו ישירות.

לדוגמה:

- מקלידים הוד-עמי או הוצאת הוד-עמי או ספרי מחשבים ומגיעים ישירות לאתר הוצאת הוד-עמי לספרי מחשבים.
- מקלידים בנק דיסקונט - ומגיעים ישירות לאתר של בנק דיסקונט.
- מקלידים בנק - ומגיעים לדף ובו רשימה של קישורים לאתרים שקוטלגו תחת המילה בנק.
- מקלידים סלקום או 052 - ומגיעים ישירות לאתר של חברת סלקום;
- מקלידים עכבר העיר - ומגיעים ישירות לאתר של העכבר;
- מקלידים 144 - ומגיעים למודיעין 144 של בזק;
- מקלידים דפי זהב, זהב דפי, מדריך דפי וכו' - ומגיעים ישירות לאתר של מדריך דפי זהב.

התחברות למערכת הניתוב החדשה של ישראל

כל שעליכם לעשות כדי להתחבר ל-NETEX הוא להתקין תוכנה קלה וחכמה:

1. יש לסגור את כל התוכנות הפתוחות כולל הדפדפנים הפועלים.
2. מתוך **סייר Windows** הפעילו את הקובץ **netex100.exe** שבתיקיה **X:\Software\NetEx**.
3. פעלו בהתאם להוראות.

בזמן ההתקנה התוכנה מזהה את הדפדפן/ים שמותקנים במחשב, והיא תפעל עם כולם. מייד אחרי סיום ההתקנה תוכלו להתחיל לגלוש חכם ובעברית.

המערכת שקופה למשתמש, כלומר היא "מתלבשת" על הדפדפן הרגיל ואינה נראית כלל. אין צורך להפעיל אותה או לבצע פעולה כלשהי כדי להשתמש בה: פשוט מקלידים את שם האתר המבוקש בשדה הכתובת של הדפדפן, ומגיעים ישר אליו.

התוכנה אינה מפריעה לעבודה רגילה עם הדפדפן. היא נכנסת לפעולה רק כאשר מקלידים נתון שאינו כתובת אינטרנט רגילה (URL). כאשר תקלידו **www.hod-ami.co.il** תגלשו ישירות לאתר **הוצאת הוד-עמי**, בדיוק כפי שנהגתם לגלוש לפני התקנת התוכנה (ללא מעורבות המערכת). אך אם תרצו, תוכלו להקליד **הוד-עמי** בשדה כתובת ולהגיע במהירות לאותו אתר בדיוק.

היכן נמצאים הקבצים הקשורים לספר זה?

התיקיה הרלוונטית לספר זה: 59240

תחת תיקיה **X:\Books\59240** תמצא קובץ ו-3 תיקיות משנה:

- תיקיה בשם **Icons** בה נמצאים קבצי ico. בהם תשתמש בפיתוח הפרויקטים השונים. העתק תיקיה זו לכונן הקשיח אצלך.
- תיקיה בשם **Images** בה נמצאים קבצי bmp. תמונות בהן תשתמש בפיתוח הפרויקטים השונים. העתק תיקיה זו לכונן הקשיח אצלך.
- תיקיה בשם **VBProjects** המחולקת לעשר תיקיות נוספות בשם **Part X** (Part 1, Part 2, Part 3 ...), תיקיה לכל פרק. בתיקיות אלו תמצא את הפרויקים המוכנים. אין צורך להעתיקם לדיסק.
- קובץ גופן בשם **wingdng2.ttf** המשמש בפרויקטים מסוימים. להתקנתו, פתח את **לוח הבקרה**, בחר **גופנים**. בחלון שייפתח פתח את תפריט **קובץ** ובחר **התקנת גופן חדש**. בתיבת הדו-שיח שתיפתח בחר את כונן התקליטורים, בחר את התיקיה (**X:\Books\59240**) בחר את הגופן ולחץ **אישור**.

אנו מבקשים להתייחס **רק** לקבצים הנמצאים בתיקיה הרלוונטית לספר. בתיקיות אחרות נמצאים קבצים הרלוונטיים לספרים אחרים של ההוצאה.

העתקת התיקיות לדיסק

כדי להעתיק את התיקיות לדיסק:

1. לחץ על לחצן **התחל**, **תוכניות**, **סייר Windows**.
2. בכונן הקשיח שלך צור תיקיה חדשה בשם **VBProjects**.
3. הצג את תוכן תיקיה **Books** אשר בתקליטור.
4. סמן את התיקיות הרצויות וגרור אותן לתיקיה שיצרת.

מעתה, בעת התייחסות בגוף הספר לתיקיות שבתקליטור, עבור לתיקיה שיצרת.

מכיון שמקור הקבצים הוא התקליטור, הם מסומנים לקריאה בלבד. רצוי לשנות מאפיין זה בדרך זו:

1. לחץ על לחצן **התחל**, **תוכניות**, **הפנייה ל-MS-DOS**.
2. בחלון DOS רשום: **attrib -r +a c:\ VBProjects *.* /s**.
3. אם העתקת את התיקיות לתיקיה אחרת, רשום זאת במקום C:**VBProjects**.

אפשרות אחרת:

1. דרך הסייר היכנס לתיקיה בדיסק, שבה נמצאים הקבצים שהעתקת.
2. סמן קובץ מסוים או את כולם על ידי Ctrl+A.
3. הצבע את סמן העכבר מעל האזור המסומן ולחץ לחיצה ימנית בעכבר.
4. מהתפריט המקוצר בחר **מאפיינים**.
5. בטל את הסימון בתיבה **קריאה בלבד** (דאג שתיבה זו תהיה ריקה).
6. לחץ על **החל**.
7. לחץ על **אישור**.

הערה!

בהחלט ייתכן שבתיקיה Software תמצא מספר תוכנות המוזכרות בספר, או שתהיינה שימושיות עבורך, למשל, תוכנת WinAmp להשמעת קבצי MP3 או MapEdit ליצירת מפת תמונה וכדומה.

התקנת Visual Basic 6 Working Model

על התקליטור שמצורף לספר, נמצאים קבצי ההתקנה של Microsoft Visual Basic 6.0 (Working Model). גירסה זו אמנם אינה מלאה, אך שימושית בהחלט.

התקנת Visual Basic 6 Working Model דורשת כי דפדפן האינטרנט Internet Explorer 4 יהיה מותקן במחשב. לצורך התקנת Visual Basic 6 Working Model עליך **להסיר** את התקנת Internet Explorer גירסה 3, אם קיימת, ורק אז לבצע את התקנת Visual Basic 6 Working Model.

הערה!

התקנת התוכנה הנבדקה בהצלחה **הוד-עמי** על מחשב שיש בו Internet Explorer 4. ההתקנה לא נבדקה על מחשב שיש בו גירסה חדשה יותר.

להתקנת Internet Explorer 4 (גירסה מלאה הכוללת תמיכה מלאה בעברית):

1. לחץ על **התחל**.
2. בחר **הפעלה**.
3. הקלד את הנתיב X:\SoftWare\IE401SR1\I386\Setup (כאשר X היא האות המייצגת את כונן התקליטורים במחשב שלך).
4. לחץ על **אישור** ועקוב אחר ההוראות המופיעות על המסך.
5. בסיום ההתקנה תתבקש לאתחל את המחשב מחדש. עשה זאת.

לתשומת לבך!

אם במחשב היתה מותקנת גירסה קודמת של Internet Explorer, במהלך ההתקנה צפויה להופיע תיבת דו-שיח המורה על התנגשות בין גרסאות. לשאלה האם ברצונך לשמור את הקובץ הקיים במחשבך השב **כן** וההתקנה תמשיך כרגיל.

לאחר התקנת Internet Explorer 4 ניתן להתקין את Visual Basic 6 Working Model באופן הבא:

1. לחץ על **התחל**.
2. בחר **הפעלה**.
3. לחץ על **עיון**.
4. עבור לכונן התקליטורים.
5. פתח את התיקיה **SoftWare**.
6. פתח את התיקיה **Visual Basic 6 Working Model**.
7. בחר בקובץ **Setup.exe** ולחץ על **פתח**.
8. בתיבת הדו-שיח **הפעלה** לחץ על **אישור** וההתקנה תחל.
9. במסך הראשון: **Visual Basic 6 Working Model** לחץ על **Next**.
10. במסך השני: **End User License Agreement** בחר: **I accept the agreement** ולחץ על **Next**.
11. במסך השלישי: **Product Number and user ID** הקלד את שמך ולחץ על **Next**.
12. במסך **Install DCOM98** ודא שתיבת הסימון מסומנת ולחץ על **Next**.
13. לאחר התקנת הרכיב **DCOM98** תוכנית ההתקנה תודיע שכעת יש לאתחל את המחשב מחדש לחץ על **OK**.
14. לאחר שהמחשב יעלה מחדש, תימשך ההתקנה. המסך הבא שיופיע הוא **Choose Common Install Folder**. בדוק אם המיקום והנתיב שמציעה תוכנית ההתקנה מתאימים לך. אם לא, לחץ על **Browse** ובחר מיקום אחר. כעת כשהמיקום והנתיב מתאימים, לחץ על **Next**.
15. במסך: **Visual Basic 6 Working Model Setup** תופיע הודעה שיש לסגור כל יישום פתוח (למעט תוכנית ההתקנה), עשה זאת (באמצעות לחיצה על לחצן היישום שבשורת המשימות) ולחץ על **Continue**.
16. בשלב הבא לחץ על **OK** והמתן בעת שתוכנית ההתקנה מחפשת רכיבים מותקנים.
17. במסך הבא שיופיע בחר בסוג ההתקנה. אם יש לך מספיק מקום בדיסק שבחרת בחר בהתקנת **Typical**.
18. אם תוכנית ההתקנה תבקש להחליף Dll ענה **No** ואחר כשתשאל אם ברצונך להשאיר את הקבצים כמות שהם ענה **Yes**.
19. בשלב זה תתבקש לאתחל את Windows מחדש, לחץ על **Restart Windows**.
20. כשהמחשב "יעלה" מחדש תופיע תיבת דו-שיח המאפשרת רישום עותק התוכנה באמצעות האינטרנט. פעל לפי ראות עיניך.

תיקית ראשית SoftWare (רשימה חלקית)

שם תוכנה	תיאור	קובץ הפעלה	מבצע
Adobe Acrobat	תוכנה לצפייה בקבצי pdf	Ar32e301.exe	התקנה
Adobe Photoshop 5 ME	תוכנה לעיבוד תמונה	Setup.exe	התקנה
Babylon	תוכנה לתרגום ולבדיקת איות	B30502h3.exe	התקנה
ClnSys	מחיקת קבצי dll שאין צורך בהם (מאמר בנושא הופיע במחשבים חלונות גליון 22)	ClnSys16.exe	התקנה
FontsPekan	גופנים בעברית	FontsPekan.exe	פריסה
Gohst 5.1c	תוכנה להעתקת תוכן כוננים קשיחים	g51c_trl.exe	פריסה
ICQ	תוכנה לתקשורת אישית באינטרנט	ICQ99a.exe	התקנה
Paint Shop Pro 5.01	תוכנה ליצירה, לעיצוב ולעיבוד תמונות	Psp501ev.exe	התקנה
Power Toys	תוכניות שירות עבור Windows 95	PowerToy.exe	פריסה
WinAmp	תוכנה להשמעת קבצי MP3 (מוסיקה)	WinAmp.exe	התקנה
WinZip	תוכנית לפריסה ולדחיסה של קבצים	WinZip7sr1.exe	התקנה
WordView	תוכנית לצפייה בקבצי doc	WordView.exe	התקנה